



# Silk Performance Manager 19.0

[Help](#)

**Micro Focus**  
**The Lawn**  
**22-30 Old Bath Road**  
**Newbury, Berkshire RG14 1QN**  
**UK**  
<http://www.microfocus.com>

**Copyright © Micro Focus 2001-2018. All rights reserved.**

**MICRO FOCUS, the Micro Focus logo and Silk Performance Manager are trademarks or registered trademarks of Micro Focus IP Development Limited or its subsidiaries or affiliated companies in the United States, United Kingdom and other countries.**

**All other marks are the property of their respective owners.**

**2018-06-06**

# Contents

<b>Introduction</b>	<b>6</b>
Product Overview	6
Silk Performance Manager Concepts	7
Pre-installed Silk Performance Manager Monitors	8
Essentials	8
Where to Distribute Monitors	9
User Roles	9
Working with Silk Performer	10
Web Services API	11
<b>Configuring Silk Performance Manager</b>	<b>13</b>
Overview	13
Setting up Monitors	14
Example	14
Adding Monitors	15
Editing Monitors	17
Deleting Monitors	17
GUI-Based Monitoring with Silk Test	18
Prerequisites for Running Silk Test Monitors	18
The Silk Test Recovery System	18
Exporting Silk Test Projects	19
Creating Silk Test Monitors	19
Scheduling Silk Test Monitors	20
Starting the Execution Server as Windows Process	20
Configuring Static Boundaries for Performance Values	21
Project Schedules	22
Time Zones	22
Configuring Project Schedules	23
Custom Monitor Schedules	23
Adding Custom Monitor Schedules	23
Editing Custom Monitor Schedules	24
Deleting Custom Monitor Schedules	25
Schedule Exclusions	25
Adding Exclusions	25
Editing Exclusions	26
Deleting Exclusions	26
Definite Runs	26
Adding Definite Runs	26
Editing Definite Runs	27
Deleting Definite Runs	27
Transaction Conditions	27
Adding Transaction Conditions	27
Editing Transaction Conditions	28
Deleting Transaction Conditions	28
Rules	29
Adding Rules	29
Editing Rules	31
Deleting Rules	32
Configuring Rule Actions	32
Custom Incidents	33
Adding Custom Incidents	34
Editing Custom Incidents	35

Deleting Custom Incidents .....	35
<b>Analyzing Results .....</b>	<b>37</b>
Overview .....	37
Analyzing Client and Infrastructure Health .....	39
Interpreting Health Detail Reports .....	40
Project Overview Reports .....	41
Project Overview Report Page .....	41
Customizing Your Start Page .....	42
Emailing Reports .....	43
Contacting Project Owners .....	44
Interpreting Measured Data .....	44
Availability .....	44
Accuracy .....	45
Performance .....	47
Performance Detail Charts .....	48
Correlating Results .....	50
Drilling Down Through Data .....	52
Downloading TrueLog Files .....	54
Incidents Log .....	54
Service Target Log .....	55
Execution Log .....	56
<b>Creating Scripts for Silk Performance Manager .....</b>	<b>58</b>
Building Reusable Monitors .....	58
Configuring Infrastructure Monitors .....	59
Requirements for Monitoring Windows Machines .....	59
Creating Silk Performance Manager Infrastructure Monitors .....	60
Maintaining Monitors .....	61
Influencing Performance Rate Calculations .....	62
Writing Result Files .....	63
Writing Action Essentials .....	63
<b>Calculating Health .....</b>	<b>65</b>
Health Rates .....	65
Health Dimensions .....	66
Overall Health .....	71
Conclusion .....	71
<b>Reports .....</b>	<b>72</b>
Creating Reports .....	72
Creating New Reports .....	72
SQL Functions for Custom Reports .....	73
Writing Advanced Queries with SQL .....	74
Sample Report .....	75
Working with Sub-Reports .....	76
Adding Sub-Reports .....	76
Deleting Sub-Reports .....	76
Report Templates .....	76
Uploading Report Templates .....	76
Downloading Report Templates .....	77
Customizing Report Templates .....	77
Removing Report Templates .....	78
Editing Report Properties .....	78
Editing Report Parameters .....	78
Working with Charts .....	79
Displaying Charts .....	79
Printing Charts .....	80
Removing Charts .....	80

Viewing Reports .....80  
Saving Reports .....81

# Introduction

This guide provides the information you need to configure Silk Performance Manager's monitoring infrastructure for effective management of your application's performance and reliability. It also shows you how to interpret and filter monitoring results using Performance Manager's reporting features.

Whether used individually or in combination, Silk products support continuous and efficient quality assurance for your organization.

## Product Overview

Silk Performance Manager is a Web-based enterprise-monitoring product that allows you to manage the performance and reliability of your applications. It is based on the latest Silk Performer technology.

### Silk Performance Manager

*Monitors* deliver "health" information for applications and can be configured to send alarms when health rates fall below specified levels. The root causes of performance, accuracy and availability issues can be identified using Performance Manager's user friendly drilldown interface.

Performance Manager helps users implement complex performance and functional transaction monitoring. It offers support for client-side business transaction monitoring of enterprise applications that are based on a wide range of technologies, including Web/HTML, client/server databases, J2EE, .NET, Web services, and ERP/CRM. It lets users define and schedule monitors distributed around the globe to measure site health based on server metrics and end-user experience metrics such as availability, accuracy and performance. Monitoring can be maintained on an ongoing basis across all tiers of an application, with data reported back in a single, intuitive interface.

Real-time reporting of collected data helps users identify performance and functional issues within production environments and is vital for trend analysis and capacity planning. Performance Manager's configurable alarm notification system enables immediate alerting of operations personnel when application performance falls below defined threshold levels. Powerful notification features such as email, pager notification, SNMP traps, and SMS messages can also be configured.

### Application Performance Management

Revenue loss, diminished productivity and damage to reputation are only some of the problems that can arise from system outages and poor e-business application response time. With Silk Performance Manager, enterprises can move toward a holistic approach to quality assurance. By extending quality assurance beyond development and into production, companies can confront challenges in a meaningful way. Functional, scalability, and performance testing-test types that are required during development to achieve software readiness-can be sustained after deployment through effective *application performance management*.

Application performance management extends quality assurance into production, thereby reducing costly downtime.

### Licensing

Performance Manager is licensed by execution server. A license must be purchased for each execution server concurrently in use.

# Silk Performance Manager Concepts

## Monitors

Monitors are entities that are used to collect system data. Synthetic monitors have one or more simulated user transactions or resource checks associated with them. They forward received data along to execution servers. Some monitors correlate directly with individual Silk Performer projects in a "one-to-many" relationship (the user, transaction, and script attributes of a single Silk Performer project can be tweaked to create multiple monitors with different attributes). Some monitors, such as simple URL checkers, are not associated with Silk Performer projects.

## Execution Servers

Execution servers are the software entities that execute Performance Manager monitors.

## Locations

Locations are the physical sites where execution servers are housed. When multiple execution servers are present at one physical location, the Performance Manager engine alternates monitoring between the servers based on load balancing algorithms. Monitors are not tied exclusively to individual locations (and by extension not to individual agents). They can be run from multiple locations simultaneously. Transactions are scheduled based on locations, not on individual execution servers, so the number of available execution servers at a location does not affect the number of times that a transaction is run at that location.

Locations are logical containers for execution servers. Since Performance Manager supports worldwide distribution PoPs (Points of Presence)-meaning, the global distribution of execution servers-it is desirable to group execution servers into locations.

## Transactions

Transactions simulate client-side business processes. Transactions may be as complex as ordering a book from a web shop, or as simple as opening a Web page. Transactions can (and normally do) have multiple measure points, such as measures for individual page requests in the case of Web business transactions.

## Incidents

Incidents are comparable to error conditions. Incidents are raised when predefined conditions are met (for example, when performance drops below a certain minimum threshold for a certain length of time and a service target violation is triggered).

## Conditions

Conditions define how monitor results are evaluated. They specify the measurements that are relevant to testing. When specified conditions are met at a certain frequency during testing (for example, when a specified threshold is exceeded 25% of the time) an incident is raised. Conditions are the building blocks from which rules are built. Examples include, "accuracy < 100%" and "availability < 100%".

## Rules

Rules define how errors are raised and what actions Performance Manager performs when incidents occur. Certain incidents may trigger email notifications while other incidents may trigger more advanced actions, such as data captures or calls to mobile phone providers.

## Exclusions

Exclusions are scheduled periods of monitor down-time. Typically exclusions are scheduled for periods of system maintenance so that anticipated system irregularities will not be factored into monitoring results. Multiple independent exclusions may be scheduled.

## Projects

Projects serve as containers for related sets of tasks and results (for example, monitors configured for a common system). Resources such as project managers and analysts are assigned to projects. Projects can only be created and maintained by system- and domain administrators.

# Pre-installed Silk Performance Manager Monitors

Performance Manager is shipped with several pre-installed monitors, which are available on the **Administration > Files > Essentials** tab.

## Client Monitors

*Client monitors* simulate end-users, or software clients, by generating actual traffic against servers. Availability, accuracy and performance are key metrics that indicate application health from the end-user perspective. Client-side monitoring involves creating simulated traffic that mimics business transactions consisting of complete end-user click-path scenarios, sequences of multi-protocol client/server interactions, and possibly sets of database interactions.

Client monitoring performs actual tasks against server systems to gather performance metrics at regular intervals from various locations.

## Infrastructure Monitors

*Infrastructure monitoring* uses both standard and custom interfaces to gather statistics regarding the state and health of systems. Typical protocols used in infrastructure monitoring for the gathering of metrics such as CPU usage and application specific statistics are SNMP, RStat and PerfMon.

# Essentials

Essentials are pre-installed monitors that ship with Performance Manager (for example, pinger, URL checker, network diagnostic monitors).

Essentials are available to all Performance Manager users, however they cannot be edited. Administrators can create custom Essentials and make them available to users.

## Silk Performance Manager Self Monitoring

Silk Performance Manager self-monitoring monitors are designed to monitor other Performance Manager server infrastructures.

- **End-User Monitor:** A Performance Manager End-User Monitor simulates a Performance Manager analyst. This monitor type tests the availability of the front-end as well as the application server from the user perspective.
- **System Health:** Provides a performance "self-monitoring" ability to Performance Manager. This Essential only requires the host name of the application server (the actual host name, "localhost" does not work). The gathered data is similar to the data on the current **System Health** page, but it can be used for actions like raising incidents or sending notifications.



**Note:** You must install a current JRE on the execution server that will monitor the application server before installing the Performance Manager execution server software.

## Pingers

This type of monitoring allows you to test a server's viability by sending it various types of echo requests. The resulting data tells you if the server is responding to the requests of users and if it is responding in a reasonable time frame. The following pinger monitors are pre-installed:

- *Pinger*: This monitor checks the availability of a specific host by sending it an Internet Control Message Protocol (ICMP) echo request via TCP/IP.
- *FtpPinger*: This monitor checks the availability of a FTP server by connecting to it and retrieving the current directory of the server.
- *LdapPinger*: This monitor checks the availability of an LDAP server by connecting to it and performing an anonymous login.
- *POP3Pinger*: This monitor connects to a POP3 server to check its availability.
- *SmtPinger*: This monitor connects to an SMTP server to check its availability.

## URL Checker

URL-checker monitors check URL's by downloading their pages and embedded objects. Verification can be defined for title and content.

## Silk Test GUI-Level Monitoring

If your application cannot be monitored with a Silk Performer-based script, use this monitor type to monitor applications via a GUI-based solution with Silk Test.

# Where to Distribute Monitors

Client monitors simulate an end-to-end user experience and evaluate whether or not applications operate as expected-covering performance, accuracy and availability. This makes active monitors well suited to service target monitoring.

One useful location for active monitoring is directly in front of firewalls, covering all local infrastructure while eliminating Internet and ISP issues. This provides accurate metrics regarding optimum application performance and is often used for WAN and end-user independent service targets.

End-users use different types of modems, have different browser versions and work from diverse locations around the world. To gather response time measurements that accurately reflect end-user perspective (including Internet speed and latency) it is crucial to run the same active monitors from multiple POPs that are distributed at locations around the world. POPs should be located where the majority of end-users are located and cover all major ISPs. For globally distributed applications, or applications that are enhanced by global-caching mechanisms, POPs should be placed in such a way that all replicated application pieces are hit.

# User Roles

The concept of user roles is important when working with Silk Performance Manager because tasks are assigned to designated groups of users.

There are five predefined user roles. These roles cannot be modified or deleted. They can however be copied and thereby used as the basis for customized roles.

## SuperUser

The SuperUser role is a special role that is granted all privileges across Performance Manager.

## Administrator

Administrator tasks include the configuring of application-, Web-, and chart-server locations; setting up and maintaining repositories and notification settings; creating accounts; configuring locations and execution servers, and others. Administrators have full access to the **Administration** area.

## Project Manager

Project Managers maintain the projects for which they are responsible. Project Managers do not have write access to the Performance Manager **Administration** unit. Project Managers can only access the projects to which they have been assigned as Project Managers, where they have full write access to all project-related features.

## Analyst

Analysts analyze the results of projects that have been assigned to them. They cannot modify project settings or schedules and have read-only privileges.

## Reporter

In addition to having all the rights of Analysts, Reporters additionally have the right to edit and delete reports in *Advanced mode*. Advanced mode allows reporters to enter, modify, and delete SQL statements for advanced reports.

# Working with Silk Performer

The integration of Silk Performance Manager and Silk Performer allows users to re-purpose Silk Performer scripts as real-time transaction monitors. This means that test cases developed during pre-production using Silk Performer can be re-used as real-time monitors of business transactions in production environments.

This guide assumes that readers are familiar with Silk Performer's features. See the Silk Performer documentation for detailed information regarding Silk Performer.

## Project attributes

Performance Manager project attributes are drawn directly from uploaded Silk Performer project attributes (those attributes that are entered on Silk Performer's **Project Attributes Configuration** page). These attributes act as customizable variables for Performance Manager projects.

See [Setting up Monitors](#) for information regarding configuring Performance Manager project attributes and uploading Silk Performer projects to Performance Manager.



**Note:** Neither Silk Performer nor Performance Manager projects must have attributes configured for them. In such instances monitors do not have **Configure Monitor - Configure Project Attributes** pages.

## TrueLog files

*TrueLog* files track complete transaction histories. When Performance Manager detects a problem with an application's health, the corresponding TrueLog can be used to detect the root cause of the problem. When uploading a monitor to Performance Manager you specify under which conditions Performance Manager should generate TrueLogs for that monitor.

See [Setting up Monitors](#) for information regarding how TrueLog should be generated for newly configured monitors.

See the Silk Performer documentation for general information regarding TrueLog technology.

## MeasureSetBound functions

Performance Manager offers two methods of calculating performance ratings for transactions. The first method uses dynamically calculated boundaries that are based on historic performance values. With this approach it is not required that you define boundaries for performance values in Silk Performer scripts, as they are calculated automatically by Performance Manager.

The second method uses two static boundaries for performance values. Outside these boundaries performance ratings are considered to be either 100% or 0%. Between these boundaries performance ratings are calculated using a logarithmic function.

With Performance Manager's Boundary Editor, administrators can adjust the static boundaries that are used for health calculation and specify which dimension each specific result is to influence. It is not necessary to define performance boundaries by manually editing Silk Performer `MeasureSetBound` script functions.

See [Boundary Editor](#) and [Calculating Health](#) for further details.

## Uploading Silk Performer projects

Silk Performer projects must be uploaded to Performance Manager before they can be utilized as monitors.

Uploading a Silk Performer project to Performance Manager from Silk Performer:

1. From Silk Performer, open the project to be uploaded and select **File > Upload Project** or select the **Deploy Monitor** workflow button and choose **Upload to Silk Performance Manager**.
2. Specify the Performance Manager address.

Uploading a Silk Performer project from Performance Manager:

1. Create a zip archive of a project (either manually or via **File > Export Project** from Silk Performer).
2. From Performance Manager, select **Administration > Files > File Pool**.
3. Click the **Upload from Browser** link. Browse to the zip archive to be uploaded and click the **Upload** link.

## Monitoring scripts

For scripting guidelines related to monitoring scripts, refer to [Creating Scripts for Silk Performance Manager](#).

# Web Services API

With its web services API, Silk Performance Manager offers a standard means of accessing the results repository.

The interface is designed as an industry-standard Web service. One significant benefit to customers who require access beyond the Performance Manager user interface is that the API will remain consistent and compatible across future versions of Performance Manager, regardless of changes to the underlying database scheme. In addition, the API provides data views that go beyond what can be offered by single SQL statements (for example, cross-project queries).

A web service (HTTP/Get) interface has been built on top of the results repository. Customers do not need to interact directly with the results repository; they can interact with the API, which will remain compatible across major version changes. See the *Silk Performance Manager API Help* for full details.

Available services include:

- `sccsystem` (System management, for example Login)
- `sccenties` (General SCA configuration data)
- `sccadminctrl` (Basic system control)

- `sventities` (Access to various Performance Manager entities)
- `svdata` (Access to the Performance Manager repository)
- `svmonconfctrl` (General Performance Manager monitor configuration)

# Configuring Silk Performance Manager

This section explains typical project manager tasks that are required for configuring a Performance Manager monitoring system, including the set up of monitors, rules, conditions, and custom incidents.

## Overview

Client monitors, at the most basic level, simulate user actions against online systems and record the effects that those actions have on system health.

As a project manager you must carefully consider the characteristics and distribution of monitors that will most help you meet your organization's needs. This section will guide you in defining the most vital aspects of your simulation infrastructure: monitors, conditions, rules, and custom incidents.

### The Projects page

The **Projects** page (**Performance Manager > Projects > Overview**) is a read-only informational page that offers important information regarding the projects that have been assigned to you. Project details include:

- **Status** - Whether a project is currently active or inactive
- **# Monitors** - The number of monitors associated with a project
- **# Transactions** - The number of transactions associated with a project
- **# Rules** - How many rules are associated with a project
- **Created by** - Who created a project
- **Created at** - When a project was created

### The Monitors tab

In addition to offering you access to monitor settings (by clicking monitor names), the **Monitors** tab (**Performance Manager > Configuration > Monitors**) offers other important functionality.

Read-only monitoring details include:

- **Transactions** - The number of transactions associated with a monitor
- **Next Run** - The next scheduled monitor execution

Available functionality includes:

- **Status** - Toggles monitor status between **Active** and **Inactive**.

### Monitor status

To change a monitor's status:

1. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click a monitor's **Active/Inactive** status link.
2. The **Deactivate Monitor Confirmation** page appears.
3. Click **Deactivate** to deactivate the monitor's status.

- **Bounds** - Enables you to change the thresholds that are used for health calculation and specify which dimension each specific result is to influence. See [Boundary Editor](#) for details.
- **Schedule** - Allows for a custom monitor schedule rather than a project-wide schedule. See [Project Schedules](#) for details.
- **Run Now** - A schedule override feature that executes monitors immediately, regardless of the next scheduled run.

## Run monitor immediately

To execute a monitor immediately, from the **Monitors** tab, click **Run Now**.

**Download:** Allows you to download the Silk Performer script packages upon which monitors are built. This is required when you need to locate and edit the latest versions of user scripts to modify simulated user behavior. Available downloads are indicated by disk buttons. Downloads that are either unavailable or available only to system administrators (for example, download packages that are associated with Essentials) are grayed out.



**Note:** Once you have edited a script in Silk Performer, upload it to Performance Manager using Silk Performer's **Upload Project** menu or the **Deploy Monitor** workflow button and reassign the uploaded project to the monitor using the **Replace Package** link on the **Configure Monitor - Define Monitor Settings** page. See [Editing Monitors](#) for more information.

To download a Silk Performer script package for a monitor:

1. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click the **Download Monitor** button of the monitor.
2. A download dialog appears. Click **Open** to open the package with Silk Performer. Click **Save** to save the package to your hard drive.

## Setting up Monitors

Client monitors simulate end-user transactions to mimic real world application usage. Monitors are derived from Silk Performer projects.

For detailed information on creating Silk Test monitors for GUI-level testing, please refer to [GUI-Based Monitoring with Silk Test](#).

## Example

This example illustrates the relationship between Performance Manager monitors and transactions; and Silk Performer projects, scripts, user groups, and transactions



**Note:** Not all monitors are built on Silk Performer scripts (for example, Silk Test scripts, simple URL checkers).

The example monitor contains the following assets:

- Silk Performer project: `project1.ltz`
- This project includes the following scripts: `script1.bdf`, `script2.bdf`
- `script1.bdf` has the following Silk Performer user groups: `vuser1`, `vuser2`
- `vuser1` has the following Silk Performer transactions associated with it: `trans1`, `trans2`

`script1.bdf:`

```
dcluser
user
vuser1
transactions
trans1 : 1;
trans2 : 1;
user
vuser2
transactions
trans3 : 1;
```

A Silk Test monitor specifies:

- Silk Performer project: `project1.ltz`

- Silk Performer script: script1.bdf
- Silk Performer user group: vuser1
- Silk Performer transactions: trans1 and/or trans2 (equates to Silk Test transaction trans1 and/or trans2)

A second Silk Test monitor might specify:

- Silk Performer project: project1.ltz
- Silk Performer script: script1.bdf
- Silk Performer user group: vuser2
- Silk Performer transactions: trans3 (equates to Silk Test transaction trans3)

A monitor for a simple e-commerce site might consist of three simulated user transactions:

- A first transaction simulating a user purchasing a product
- A second transaction simulating a user who searches for a product, but does not make a purchase
- A third transaction might simply confirm the availability and accuracy of the home page.

Taken together three such transactions could effectively test a system's health. A monitor for a more complex system would require more transactions.

## Adding Monitors

Your system administrator must create a Performance Manager project before you can create a monitor.

To add a new monitor:

1. On the **Projects** page (**Performance Manager > Projects**), click the project to which you want to add a monitor.
2. Select the **Performance Manager > Configuration > Monitors** tab. The **Monitors** tab displays all monitors that are currently selected for the project. Click **Add New Monitor**.
3. The **Configure Monitor - Select Monitor Type** page contains a list of all monitors that have been uploaded to Performance Manager from Silk Performer, including pre-installed monitors (Essentials). Click a monitor upon which you want your new monitor to be based.
4. (Silk Performer script monitors only): The **Configure Monitor - Select Script** page contains all scripts that comprise the selected monitor type. Select a script to use it as a monitor.
5. Click **Next**.



**Note:** If your Silk Performer project contains only one script, the **Configure Monitor - Select Script** page does not appear.

6. (Silk Performer script monitors only): The **Configure Monitor - Select User Group** page contains all the user groups that associated with the project. Select the user group that is to be included in the monitor.
7. Click **Next**.



**Note:** If your Silk Performer script contains only a single user group, the **Configure Monitor - Select User Group** page does not appear.

8. (Silk Performer script monitors only): Select the transactions that are to be included in the monitor from the **Configure Monitor - Select Transactions** page.
9. Click **Next**.



**Note:** With Silk Performer script monitors, if your Silk Performer script contains only one transaction the **Configure Monitor - Select Transactions** page does not appear.

10. If your monitor is built on a Silk Performer script that contains project attributes, the **Configure Monitor - Customize Project Attributes** page appears.



**Note:** Performance Manager project attributes are drawn directly from uploaded Silk Performer project attributes. See [Working with Silk Performer](#) for details.

11. Click **Next**. The **Configure Monitor - Define Monitor Settings** page allows you to adjust the attributes of the selected script so that it serves as an effective monitor. With Performance Manager, appropriate monitor configuration pages are provided automatically. The available fields on the **Configure Monitor - Define Monitor Settings** page vary depending on the monitor type you have based the new monitor on.
12. Enter a name for the monitor in the **Monitor Name** field.
13. Select a profile for the monitor from the **Profile** list. Note that profiles come directly from Silk Performer project profile settings, and may not be included. In such cases the **Profile** list is not available.
14. Select a browser type from the **Browser** list, or leave the field set to default from profile to use the browser specified in the Silk Performer project profile. This feature allows you to override Silk Performer's browser profile setting for this project.
15. If you want to distinguish hits generated by Performance Manager from other traffic, select **Identify as Silk Performance Manager**. This setting will add a `Silk Performance Manager` prefix to the user agent of the HTTP header, thus allowing for separate entries in the Web statistics.



**Note:** If the monitored Web application's behavior is based on user agent information, turning **Identify as Silk Performance Manager** on may cause application misbehavior. If you do not need to differentiate the traffic, switch this feature off.

16. Select a connection speed from the **Connection Speed** list, or leave the field set to default from profile to use the connection speed specified in the Silk Performer project profile. This feature allows you to override Silk Performer's connection speed profile setting for this project.
17. Specify how Performance Manager should calculate performance rates by selecting one of the **Performance Rate Calculation** radio buttons. Have performance rates calculated **Based on automatically adjusted bounds** if you want to have Performance Manager calculate performance based on past performance levels. See [Analyzing Results](#) and [Calculating Health](#) for more information regarding performance and health calculations. Have performance rates calculated **Based on static boundaries** if you have run benchmark tests against your system and are familiar with the specific threshold boundaries against which performance should be calculated. Although Performance Manager's boundary editor is the recommended means of changing static boundaries (See [Boundary Editor](#)), an alternate method that you may find useful when initially setting boundaries involves `MeasureSetBounds` scripting functions.

Here is an example `MeasureSetBounds` function:

```
function InitMeasureBounds
begin
    MeasureSetBound (NULL, MEASURE_PAGE_PAGETIME, 1, 1.0);
    MeasureSetBound (NULL, MEASURE_PAGE_PAGETIME, 2, 2.0);
    MeasureSetBound (NULL, MEASURE_TRANS_TRANSBUSYOK, 1, 1.0);
    MeasureSetBound (NULL, MEASURE_TRANS_TRANSBUSYOK, 2, 2.0);
    MeasureSetBound ("TBUYER-019-Order Processed", MEASURE_PAGE_PAGETIME, 2,
0.637*4.0);
    MeasureSetBound ("TBUYER-018-Order Information(#1)",
MEASURE_PAGE_PAGETIME, 2, 1.521*4.0);
    MeasureSetBound ("TBUYER-017-Order Information", MEASURE_PAGE_PAGETIME,
2, 1.604*4.0);
    MeasureSetBound ("TBUYER-017-Order Information", MEASURE_PAGE_PAGETIME,
1, 1.604*1.0);
    ...
end InitMeasureBounds
```

18. Utilizing the meta information capabilities offered by Performance Manager, you can define how monitor results (**Transaction Response Times**, **Page Timers**, and **Custom Measurements**) affect overall system health. Results can be disabled using meta information. In such cases users can specify a no results or display only setting for transaction response time. Alternatively, some customers use the Silk Performer Recorder to monitor complex business transactions and design custom timers that are ideally suited to their measurement needs. In such cases Performance Manager's standard timers might alter results or make the interpretation of results confusing. A user could then specify that all **Custom Measurements** be disregarded entirely (no results) or only be displayed (display only), rather than factored into performance ratings (performance rating).

19. Though the default error-reporting setting (**Report availability and accuracy separately**) is applicable for most situations, Performance Manager enables you to optionally specify that availability and accuracy errors be reported only as availability errors (**Report all errors as availability errors**) or not recorded at all (**Do not record errors**). For example, if you are only interested in the performance data generated by a monitor, you might specify that availability and accuracy errors not be recorded at all (**Do not record errors**).



**Note:** If you select **Do not record errors**, errors will not be factored into overall system health, but they will be recorded in the execution log.

20. Specify how you want TrueLog to be generated for this monitor by selecting one of the **Generate TrueLog** radio buttons.

- Select **On Error** to have TrueLog generated only when errors are encountered. This is the recommended approach.
- Select **Always** to have TrueLog track all activity. Note that this is a processing and storage intensive option that may affect system performance.
- Select **Never** to not have TrueLog generated for this monitor.

21. Select the **Generate Default Output File (.wrt)** check box to have `.wrt` files written and accessible via Performance Manager's **Execution Log** (as TrueLog files are).



**Note:** Scripts that include `Write()` or `WriteLn()` statements end with the extension `.wrt`. If the **Generate Default Output File (.wrt)** check box is not checked, such files will not be written.

22. With **Create default rule with default condition(s) automatically** you can create rules that are activated when availability or accuracy drops below 100, or when performance reaches 0 (when a performance measure reaches bound 2). Simply select the **Availability**, **Accuracy**, and/or **Performance** check boxes to activate these rules. The default rule can later be modified in the rules section. See [Editing a Rule](#) for more details on rules.

23. Select the **Run exclusive** option box to prevent other monitors from running simultaneously on the same execution server. This is particularly useful for Silk Test monitors.

24. Click **Finish** to save the monitor with the project-wide schedule.

## Editing Monitors

To edit an existing monitor:

1. On the **Projects** page (**Performance Manager > Projects**), click the project of which you want to edit a monitor.
2. Select the **Performance Manager > Configuration > Monitors** tab. The **Monitors** tab displays all monitors that are currently selected for the project. Click the monitor you want to edit.
3. If your monitor is built on a Silk Performer script that contains project attributes, the **Configure Monitor - Customize Project Attributes** page appears.



**Note:** Performance Manager project attributes are drawn directly from uploaded Silk Performer project attributes. See [Working with Silk Performer](#) for details.

4. Click **Next**. The **Configure Monitor - Define Monitor Settings** page allows you to adjust the attributes of the selected script so that it serves as an effective monitor. With Performance Manager, appropriate monitor configuration pages are provided automatically. The available fields on the **Configure Monitor - Define Monitor Settings** page vary depending on the monitor type you have based the new monitor on. See [Adding Monitors](#) for details regarding parameters that are available for editing.

## Deleting Monitors

To delete a monitor:

1. On the **Projects** page (**Performance Manager > Projects**), click the project of which you want to delete a monitor.

2. Select the **Performance Manager > Configuration > Monitors** tab. The **Monitors** tab displays all monitors that are currently selected for the project. Click the **Delete** button of the monitor you want to delete.
3. The **Delete Monitor Confirmation** message appears. Click **Delete** to permanently delete the monitor from the project.

## GUI-Based Monitoring with Silk Test

This section introduces you to the process of integrating Silk Test GUI-based tests into Performance Manager. Both products are fully integrated and enable you to create Silk Test monitors just like creating Silk Performer monitors.

### Prerequisites for Running Silk Test Monitors

To run Silk Test monitors, the only software requirements are that Silk Test be installed on the execution servers where Silk Test monitors are scheduled to run.

For more information on prerequisites and Terminal Services settings, please refer to the Silk Performer Workbench Help (**Load Testing Specific Application Types > GUI-Level Testing Support > Configuring Windows for GUI-Level Testing**).

The majority of performance monitors configured in Performance Manager are driven by protocol-based scripts developed in Silk Performer. However some applications cannot be easily modeled with Silk Performer. These applications might use non-standard protocols or use encryptions that can not be easily scripted by Silk Performer (for example, client/server applications.) In these cases you may choose to monitor the application via a GUI-based solution with Silk Test.

Once a Silk Test monitor script has been created and configured properly in Performance Manager, the monitor will:

- Periodically start Silk Test, which will open up the application-under-test (AUT)/monitor on the execution server machine.
- Take control of the keyboard and cursor of the execution server machine.
- Physically execute the script in the application GUI.
- Capture application performance metrics using pre-defined timers.
- Close the application-under-test/monitor.
- Close Silk Test.
- Send the application performance data back to the Performance Manager application server.

### The Silk Test Recovery System

The Silk Test Recovery System is a series of functions that wrap around the SilkTest test cases.

The Recovery System is designed to allow unattended executions by ensuring that the application under test is always available in a "testable state." These "testable states" are called "Application States" in Silk Test. At the beginning of a test case the Recovery System will ensure the application under test/monitor is running. If the application is not started the Application State functionality will automatically start the test/monitor application. If the application is already running then the Recovery System will close down all windows and dialog boxes except for the main application window. At this point the actual test case will begin.

Once the test case has executed, the control in Silk Test will again return to the Recovery System. The Recovery System will once again close any extraneous dialog boxes and leave the main application window running. If there were any errors detected during the test case execution the Recovery System will report those errors to the Silk Test results.

Silk Test is delivered with a default application state that is automatically configured during project creation. This `DefaultBaseState` has generic functions for handling the automatic recovery of any application.

However there are times when the `DefaultBaseState` needs to be modified in order to handle non-standard applications or to enhance the testing process. Silk Test provides a straightforward methodology for overwriting or modifying the Recovery System. This methodology can be found in detail in the *Silk Test Help*.

## Exporting Silk Test Projects

Before you can upload a Silk Test project to Performance Manager, you need to export the project as archive.

To export a Silk Test project:

1. Select **File > Export Project**. You can only export a project if you have that desired project open.
2. On the **Export Project** dialog, enter the folder to which you want to export the project or click the **Browse** button to locate the export folder. The default location is the parent directory of the project folder, i.e., the folder containing the project file, not the project's current location.
3. Select the **Export to single Silk Test package** check box to package the Silk Test project into a single compressed file.
4. Click **OK**. Silk Test determines all the files necessary for the project and copies them to the selected directory or compresses them into a package. Silk Test displays a warning message if any of the files could not be successfully packaged and gives you the option of continuing.



**Note:** If you have a crash during the export process, we strongly suggest deleting the partially packaged project (if any), and starting the process over again.

## Creating Silk Test Monitors

In order to execute a Silk Test monitor in Performance Manager you must first have a Performance Manager project created and a suitable execution server defined (see Performance Manager Administration Guide for directions).

To create a Silk Test monitor:

1. Select a project (**Performance Manager > Projects**).
2. Select **Performance Manager > Configuration** and click the **Monitors** tab. The **Monitors** tab displays all monitors that have been created for the project. Click the **Add New Monitor** link.
3. The **Configure Monitor - Select Monitor Type** page contains a list of all monitors that have been uploaded to Performance Manager. Select **Custom Monitors > <your monitor>.stp**.
4. The **New Silk Test Monitor Settings** dialog appears. The following settings are available:
  - **Test Script:** The test script to execute. All scripts contained in the selected `stp` file are available in the list.
  - **Test Case:** The test case to execute. The test case can be selected from a list or entered manually in the **Custom** field.
  - **Test Data** (optional): The test data for the Silk Test execution. If several arguments are passed to Silk Test, they have to be separated by a comma (,). If a String argument is passed to Silk Test, the argument must be surrounded by double quotes (").
  - **Terminal Services User** (optional): The Terminal Service User credentials are specified for each execution server. Checking the **Override Execution Server Settings** check box and entering a valid **Username** and **Password** for the monitor allows overriding the execution server user credentials for this monitor.



**Note:** This setting will be used on all execution servers where the monitor is executed.

- **Color Depth:** The color depth for the Terminal Service session.
- **Time-out:** The execution time-out for the monitor execution in seconds.

5. Click **Next**. The **Configure Monitor - Define Monitor Settings** page appears, allowing you to adjust the attributes of your test script so that it serves as an effective monitor (see [Adding a Monitor](#) for details regarding parameters that are available for editing).
6. Click **Finish** to add the monitor.

## Scheduling Silk Test Monitors

### Run Once Now

Before actually scheduling a monitor execution, we recommend to try out whether everything works first. If Silk Test is open, shut it down. From the Performance Manager **ConfigurationMonitors** page, select the **Run Now** button to schedule the monitor for immediate execution.

At this point refrain from touching the mouse or keyboard. The Silk Test monitor will take over the desktop and execute the script.

### Verifying Script Execution

After the Silk Test monitor has executed, browse to the execution logs within Performance Manager. To do this go to **Reports > Execution Log**. An execution should be listed with a very recent timestamp (sometimes there is a delay in the **Execution Log** reporting). Select the **Results** button corresponding to the monitor execution.

### Results

If you have included custom timers in your Silk Test script, the custom timer results will appear on the **Results** page towards the bottom. This timer will be saved with Performance Manager and used as the data point for the monitor execution.

### Scheduling Silk Test Monitors

You schedule Silk Test monitors in exactly the same fashion as other monitors. Navigate to **Configuration > Monitors** and select the **Schedule** button. Modify the schedule as needed (see [Adding Custom Monitor Schedules](#) for detailed information).

### Using Timers

If you are going to use specific timers and you want those timers to be visible in Performance Manager, you will need to create timers in your `xxx.t` file - see the example below for creating, setting and ending the timer:

```
[ ] HTIMER Checkbox
[ ] Checkbox=TimerCreate( "Checkbox" )
[ ]
[ ] TimerStart(Checkbox)
[ ]
[ ] // wrap around the action
[ ] TestApplication.SetActive ( )
[ ] TestApplication.Control.CheckBox.Pick ( )
[ ] CheckBox1.SetActive ( )
[ ] CheckBox1.TheCheckBox.Check ( )
[ ] CheckBox1.Exit.Click ( )
[ ]
[ ] TimerStop(Checkbox)
[ ] TimerDestroy(Checkbox)
```

## Starting the Execution Server as Windows Process

Start the execution server service as a Windows process if your monitor needs to run using the credentials of the currently logged in user.

Monitors run in Windows Terminal Services sessions by default. Note that multiple Terminal Services sessions are only supported by Windows Server operating systems. Other Windows operating systems like Home or Professional editions support only limited Terminal Services sessions.

The execution server can be run either as a Windows service or a Windows process. In most instances this is preferable since it is active even when a user logs off, which means the execution server is always available, unless the computer is powered off. However, the Windows service is launched using the default system account and this may not always be suitable—for example, launching certain executables within a monitor may require particular users' credentials. In such instances it may be necessary for the execution server to be launched as a Windows process—this uses the credentials of the currently logged in user.

To start the Performance Manager execution server as a Windows process:

1. Double-click the **Silk Performance Manager Service Manager** tray icon in the Windows task bar. The *Silk Performance Manager Service Manager* displays, with up to five tabs visible, depending on the services that are installed on this computer.

2. Click the **Execution Server** tab.

This tab represents the Performance Manager execution server, running as a Windows system service.

3. Click **Stop** to stop the execution server system service.

4. Click **Query Status** to check the service's status.

Make sure that the service status is `stopped`.

5. Uncheck **Run at start-up** to prevent that the service is started after computer re-boot.

6. Click the **Execution Server (Process)** tab.

This tab represents the Performance Manager execution server, running as a Windows process.



**Note:** The Windows process is launched with the credentials of the user who is currently logged in. Make sure that this user has sufficient privileges to accomplish the tasks you are planning to execute with Performance Manager.

7. Click **Start** to start the execution server as a Windows process.

8. Check **Run at start-up** so that the process is started after computer re-boot and re-login.

9. Click **OK** to finish managing the execution server. The **Service Manager** closes, but remains active in the system tray.

## Configuring Static Boundaries for Performance Values

Using the Boundary Editor, administrators can easily change the thresholds that are used for health calculation and specify which dimension each specific result is to influence.

The Boundary Editor is used to configure static boundaries for performance values. When relying on dynamically calculated boundaries that are based on historic performance values, it is not required that you define boundaries for performance values in Silk Performer scripts—they are calculated automatically by Performance Manager. You can however use dynamic boundaries and restart the calibration process beginning with the current date. To do so, go to **Performance Manager > Configuration > Monitors** and click the **Reset bounds** icon of the monitor you want to reset the boundaries for.

With the Boundary Editor you can actually change the parameters of each result that is provided by a monitor. You can even change the names of results and specify new boundaries, all from within Performance Manager. Reasonable default values for all parameters guarantee compatibility with most sophisticated monitoring scenarios and will not complicate the monitor configuration process. Monitor context parameters even allow you to disregard page/custom timers that are not relevant to specific dimensions and configure the handling of availability/accuracy errors for monitors. See [Adding a Monitor](#) for details regarding setting up monitor parameters.

To edit bound-based health values:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to edit health values.
2. Go to **Performance Manager > Configuration > Monitors**. The **Monitors** tab displays all monitors that are currently available for the project. Click the **Edit Bounds** icon of the monitor for which you want to edit health values. The **Edit Bounds for Monitor...** page for the selected monitor appears.



**Note:** The **Edit Bounds** icon is only displayed for monitors that are based on static boundaries. To define static boundaries, you need to re-open the script in Silk Performer, go to **Project > Set Response Time Thresholds**, ensure **Generate project Attributes** is checked, click **OK**, and upload the monitor back to Performance Manager.

3. From here you can change the parameters of all results provided by the monitor. Edit the name of a counter in the **Custom Counter** column.
4. Edit boundary 1 (the "lower" boundary) in the **Bound1** column.
5. Edit boundary 2 (the "upper" boundary) in the **Bound2** column.
6. Use the pull-down menu in the **Rating** column to specify which health dimension each counter is to apply to (**Availability**, **Accuracy**, or **Performance**). Or select **Display** only to have a counter's results displayed, but not factored into health ratings.
7. Click **Save** to save your revised boundary settings.

## Project Schedules

Project schedules define when the monitors in a given project are executed.

Administrators have the option of specifying project-specific schedules when they create projects, however defining project schedules is generally considered a project manager's responsibility. In addition to being able to create and edit project-wide schedules, project managers can also create custom schedules for individual monitors; see [Custom Monitor Schedules](#) for details.

The default project schedule:

- Begins immediately
- Does not have a scheduled end
- Runs every five minutes
- Has no exclusions
- Runs on all available locations

## Time Zones

Performance Manager executes monitors over networks of execution servers. The Internet enables such networks to be spread across different time zones. Performance Manager makes time-zone handling easy by automatically saving date and time values in the local time zone defined in the user settings.

### Example:

User setting: Boston (EST = GMT-05:00)

Execution server 1: Boston (EST = GMT-05:00)

Execution server 2: San Francisco (PST = GMT-08:00)

Monitor #1 is scheduled to execute at 5:00 pm on Execution server 1 and Monitor #2 is scheduled to execute at 5:00 pm on Execution server 2.

Both monitors will execute at 5:00 PM EST, which is 2:00 PM PST in San Francisco.

Reported results for monitor #2 will show an execution time of 5:00 PM in addition to the execution server time (GMT-08:00).

# Configuring Project Schedules

To configure a project schedule:

1. On the **Projects** page (**Performance Manager > Projects**), click a project for which you want to edit the schedule.
2. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click the **Project Schedules: Clientside** link to adjust the schedule of the project's client-side monitors.
3. On the **Configure Schedule** page, specify a **From** time (month, day, year, hour, minute, second, AM/PM) when the project's monitors should begin to run.
4. Specify the **Interval** (day, hour, minute) at which the project's monitors should execute.



**Note:** When monitors are comprised of multiple transactions, transactions are executed in quick succession. Individual transactions can't be scheduled independently within monitors. Ensure that the execution interval you select provides adequate time for the monitor's transactions to execute (for example, a monitor that requires 90 seconds to complete its transactions, can not be executed at 60 second intervals).

5. Specify how long the project's monitors are to **Run** (forever, a specific number of times, or until a specific date).
6. In the **Run From** list, select at least one location from which the project's monitors are to be executed. Note that monitors can be run simultaneously from multiple locations. Use the **Ctrl** key to select multiple locations.
7. Select a **Cascaded Delay**. This setting avoids that a monitor is started simultaneously across all locations to avoid workload peaks on the system under test. Example: Setting the cascaded delay to 5 seconds for a monitor that is scheduled to run at 5:00 AM results in the monitor to run on the first location at 5:00:00 AM, on the second location at 5:00:05, and on the third location at 5:00:10.



**Note:** If the duration of the cascaded runs exceeds the execution interval, the skipped monitor execution will not be executed.

8. Click **Save** to return to the **Monitors** tab. Alternatively, you may advance to the **Add Exclusions** page to add exclusions. See [Schedule Exclusions](#) for details regarding exclusions.

## Custom Monitor Schedules

In addition to having privileges for changing schedules project-wide, project managers can also create custom schedules for individual monitors that override project-wide schedules.

## Adding Custom Monitor Schedules

To add a custom monitoring schedule for an individual monitor:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to add a custom monitor schedule.
2. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click the clock button in the **Schedule** column that corresponds to the monitor you want to schedule.  
Alternatively, from the **Configure Monitor Settings** page, click **Finish with special Schedule** to go to the **Configure Schedule** page.
3. On the **Configure Schedule** page, specify a **From** time (month, day, year, hour, minute, second, AM/PM) when the project's monitors should begin to run.
4. Specify the **Interval** (day, hour, minute) at which the project's monitors should execute.



**Note:** When monitors are comprised of multiple transactions, transactions are executed in quick succession. Individual transactions can't be scheduled independently within monitors. Ensure that the execution interval you select provides adequate time for the monitor's transactions to execute (for example, a monitor that requires 90 seconds to complete its transactions, can not be executed at 60 second intervals).

5. Specify how long the project's monitors are to **Run** (forever, a specific number of times, or until a specific date).
6. In the **Run From** list, select at least one location from which the project's monitors are to be executed. Note that monitors can be run simultaneously from multiple locations. Use the **Ctrl** key to select multiple locations.
7. Select a **Cascaded Delay**. This setting avoids that a monitor is started simultaneously across all locations to avoid workload peaks on the system under test. Example: Setting the cascaded delay to 5 seconds for a monitor that is scheduled to run at 5:00 AM results in the monitor to run on the first location at 5:00:00 AM, on the second location at 5:00:05, and on the third location at 5:00:10.



**Note:** If the duration of the cascaded runs exceeds the execution interval, the skipped monitor execution will not be executed.

8. From the **Concurrent Runs** list, specify the number of monitors that are concurrently started on the scheduled locations. This setting limits the number of monitors that are concurrently started on the scheduled locations to the configured value. Example: If a monitor is scheduled on 100 locations and **Concurrent Runs** is set to 10, the first schedule interval executes the monitor on locations 1-10, the second interval on locations 11-20, and so on.



**Note:** The underlying round-robin principle does not always arrange the locations into the same groups, depending on if the number of locations can be divided through the number of concurrent runs. Example: for 13 locations and 5 concurrent runs, the monitor will execute on locations 1,2,3,4,5 - 6,7,8,9,10 - 11,12,13,1,2 - 3,4,5,6,7, and so on.

9. Click **Save** to return to the **Monitors** tab. Alternatively, you may advance to the **Configure Schedule Exclusion** page to add exclusions or to the **Configure Definite Run** page to configure definite runs. See [Schedule Exclusions](#) and [Definite Runs](#) for detailed information.

## Editing Custom Monitor Schedules

To edit an existing custom monitoring schedule:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to edit an existing custom monitor schedule.
2. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click the clock button in the **Schedule** column that corresponds to the monitor you want to re-schedule.
3. On the **Configure Schedule** page, specify a **From** time (month, day, year, hour, minute, second, AM/PM) when the project's monitors should begin to run.
4. Specify the **Interval** (day, hour, minute) at which the project's monitors should execute.



**Note:** When monitors are comprised of multiple transactions, transactions are executed in quick succession. Individual transactions can't be scheduled independently within monitors. Ensure that the execution interval you select provides adequate time for the monitor's transactions to execute (for example, a monitor that requires 90 seconds to complete its transactions, can not be executed at 60 second intervals).

5. Specify how long the project's monitors are to **Run** (forever, a specific number of times, or until a specific date).
6. In the **Run From** list, select at least one location from which the project's monitors are to be executed. Note that monitors can be run simultaneously from multiple locations. Use the **Ctrl** key to select multiple locations.
7. Select a **Cascaded Delay**. This setting avoids that a monitor is started simultaneously across all locations to avoid workload peaks on the system under test. Example: Setting the cascaded delay to 5

seconds for a monitor that is scheduled to run at 5:00 AM results in the monitor to run on the first location at 5:00:00 AM, on the second location at 5:00:05, and on the third location at 5:00:10.



**Note:** If the duration of the cascaded runs exceeds the execution interval, the skipped monitor execution will not be executed.

8. From the **Concurrent Runs** list, specify the number of monitors that are concurrently started on the scheduled locations. This setting limits the number of monitors that are concurrently started on the scheduled locations to the configured value. Example: If a monitor is scheduled on 100 locations and **Concurrent Runs** is set to 10, the first schedule interval executes the monitor on locations 1-10, the second interval on locations 11-20, and so on.



**Note:** The underlying round-robin principle does not always arrange the locations into the same groups, depending on if the number of locations can be divided through the number of concurrent runs. Example: for 13 locations and 5 concurrent runs, the monitor will execute on locations 1,2,3,4,5 - 6,7,8,9,10 - 11,12,13,1,2 - 3,4,5,6,7, and so on.

9. Click **Save** to return to the **Monitors** tab. Alternatively, you may advance to the **Configure Schedule Exclusion** page to add exclusions or to the **Configure Definite Run** page to configure definite runs. See [Schedule Exclusions](#) and [Definite Runs](#) for detailed information.

## Deleting Custom Monitor Schedules

To delete an existing custom monitoring schedule:

1. On the **Projects** page (**Performance Manager > Projects**), click the project from which you want to delete an existing custom monitor schedule.
2. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click the **Delete** button in the **Schedule** column that corresponds to the monitor you want to delete. The **Delete Custom Schedule** confirmation message appears.
3. Click **Delete** to permanently delete the custom monitor schedule and return to the **Monitors** tab.

## Schedule Exclusions

Exclusions are weekly scheduled periods of monitor down-time.

Typically exclusions are scheduled for periods of regular system maintenance so that anticipated system irregularities will not be factored into monitoring results. Multiple exclusions can be scheduled.

## Adding Exclusions

To add a schedule exclusion:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to add a schedule exclusion.
2. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click the clock button in the **Schedule** column that corresponds to the monitor for which you want to schedule an exclusion.
3. On the **Configure Schedule** page, click **Add Exclusion**.
4. In the **Don't run on** section of the **Configure Schedule Exclusion** page, select one or more days on which you want to schedule an exclusion.
5. Select the time of day (hour, minute, second, AM/PM) when the exclusion is to begin from the **From** lists.
6. Select the time of day (hour, minute, second, AM/PM) when the exclusion is to end from the **To** lists.
7. Click **OK** to save the exclusion.

Alternatively, click **Clear** to clear your selections or **Cancel** to return to the **Configure Schedule** page.

# Editing Exclusions

To edit a scheduled exclusion:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to edit a scheduled exclusion.
2. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click the clock button in the **Schedule** column that corresponds to the monitor for which you want to edit a scheduled exclusion.
3. On the **Configure Schedule** page, click the **Edit** icon that corresponds to the exclusion you want to edit.
4. In the **Don't run on** section of the **Configure Schedule Exclusion** page, select/deselect the days on which you want to schedule an exclusion.
5. Select the time of day (hour, minute, second, AM/PM) when the exclusion is to begin from the **From** lists.
6. Select the time of day (hour, minute, second, AM/PM) when the exclusion is to end from the **To** lists.
7. Click **OK** to save your changes.

Alternatively, click **Clear** to clear your selections or **Cancel** to return to the **Configure Schedule** page.

# Deleting Exclusions

Enter the prerequisites here (optional).

Enter the context of your task here (optional).

1. On the **Projects** page (**Performance Manager > Projects**), click the project from which you want to delete a scheduled exclusion.
2. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click the clock button in the **Schedule** column that corresponds to the monitor for which you want to delete a scheduled exclusion.
3. On the **Configure Schedule** page, click the **Delete** icon that corresponds to the exclusion you want to delete.

# Definite Runs

A definite run is a specific time at which a monitor will execute, regardless of its usual schedule or any configured exclusions. Multiple definite runs can be scheduled.

# Adding Definite Runs

To add a definite run:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to add a definite run.
2. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click the clock button in the **Schedule** column that corresponds to the monitor for which you want to schedule a definite run.
3. On the **Configure Schedule** page, click **Add Definite Run**.
4. Select the day (month, day, year) and the time of day (hour, minute, second, AM/PM) when the monitor shall execute from the lists.
5. Click **OK** to save the definite run.

# Editing Definite Runs

To edit a definite run:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to edit a definite run.
2. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click the clock button in the **Schedule** column that corresponds to the monitor for which you want to edit a definite run.
3. On the **Configure Schedule** page, click the **Edit** icon that corresponds to the definite run you want to edit.
4. Select the day (month, day, year) and the time of day (hour, minute, second, AM/PM) when the monitor shall execute from the lists.
5. Click **OK** to save your changes.

# Deleting Definite Runs

To delete a definite run:

1. On the **Projects** page (**Performance Manager > Projects**), click the project from which you want to delete a definite run.
2. On the **Monitors** tab (**Performance Manager > Configuration > Monitors**), click the clock button in the **Schedule** column that corresponds to the monitor for which you want to delete a definite run.
3. On the **Configure Schedule** page, click the **Delete** icon that corresponds to the definite run you want to edit.

# Transaction Conditions

Conditions define how monitor results are evaluated.

They specify the measurements that are relevant to testing. When specified transaction conditions are met at a certain frequency during testing (for example, when a specified limit is exceeded 25% of the time) an incident is raised. Conditions are the building blocks from which rules are built. Examples include, "if accuracy is < 100%" and "if availability is > 0%".



**Note:** Transactions don't necessarily require that conditions be associated with them.

## The Conditions Page

The **Conditions** page (**Performance Manager > Configuration > Conditions**) includes a list of all currently configured conditions. It also includes high-level information about each condition, including:

- The **Condition** column, which includes a system health measurement, an operator, and a threshold comparison value (for example, `Accuracy < 100.0%`).
- The **Locations** column, which indicates the locations to which each condition applies.
- The **Transaction Scope** column, which indicates to which transactions the conditions are applied.

For information about the Add Transaction Condition button, see [Adding Transaction Conditions](#).

For information about deleting transaction conditions, see [Deleting Transaction Conditions](#).

# Adding Transaction Conditions

To add a transaction condition:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to add a transaction condition.
2. On the **Conditions** tab (**Performance Manager > Configuration > Conditions**), click **Add Transaction Condition**.
3. Select the appropriate transaction from the **Configure Condition - Select the Transaction to define a condition on** page.

The **Configure Condition** page appears. This page is populated with historical data regarding past system performance. **Average**, **Minimum**, and **Maximum** measurements are listed in the table, along with any bounds that may have been defined in the corresponding Silk Performer script package, derived from Boundary Editor settings, or are actual dynamic boundary values. The measurement **Unit** type (% , seconds, etc.) is also included.

4. Enter a name for the condition in the **Condition Name** field.
5. Select the radio button of the system health **Measurement** you want to serve as the basis of the condition.
6. Specify how the condition should interpret the measurement by selecting an operator from the **Operator** list (<, =, or >) and entering a value in the **Threshold Comparison Value** field.
7. Define the monitoring locations that are to be considered in evaluating the condition (**At all locations**, **At 75% of locations**, **At 50% of locations**, **At 25% of locations**, or **At one location**).
8. Click **Save** to save the condition parameters.

## Editing Transaction Conditions

To edit a transaction condition:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to edit a transaction condition.
2. On the **Conditions** tab (**Performance Manager > Configuration > Conditions**), select the condition you want to edit from the **Condition Name** column.
3. Select the appropriate transaction from the **Configure Condition - Select the Transaction to define a condition on** page.

The **Configure Condition** page appears. This page is populated with historical data regarding past system performance. **Average**, **Minimum**, and **Maximum** measurements are listed in the table, along with any bounds that may have been defined in the corresponding Silk Performer script package, derived from Boundary Editor settings, or are actual dynamic boundary values. The measurement **Unit** type (% , seconds, etc.) is also included.

4. Edit the name of the condition in the **Condition Name** field.
5. Select the radio button of the system health **Measurement** you want to serve as the basis of the condition.
6. Specify how the condition should interpret the measurement by selecting an operator from the **Operator** list (<, =, or >) and entering a value in the **Threshold Comparison Value** field.
7. Define the monitoring locations that are to be considered in evaluating the condition ( **At all locations**, **At 75% of locations**, **At 50% of locations**, **At 25% of locations**, or **At one location**).
8. Click **Save** to save your changes.

## Deleting Transaction Conditions

To delete a transaction condition:

1. On the **Projects** page (**Performance Manager > Projects**), click the project from which you want to delete a transaction condition.

2. On the **Conditions** tab (**Performance Manager > Configuration > Conditions**), click the **Delete** button that corresponds to the condition you want to delete. The **Delete Condition** confirmation message appears.
3. Click **Delete** to permanently delete the condition.

## Rules

Rules define how errors are raised and the actions that Performance Manager performs when incidents occur.

Certain incidents may trigger email notifications while other incidents may trigger more advanced actions known as action Essentials. Action Essentials are Essentials that act as powerful notification alarms. They can be used to execute any action that can be programmed with a Silk Performer script. Rules can also trigger the violation of Service Target Agreements. Rules are built upon conditions. Therefore you must configure conditions before you configure rules.

Conditions within a rule are evaluated from top to bottom, and the **AND** operator has a stronger binding than the **OR** operator.

### The Rules page

The **Rules** tab (**Performance Manager > Configuration > Rules**) includes a list of all currently configured rules. It also includes high-level information about each rule, including:

- The **Condition Expression** column, which details the parameters of each condition.
- The **Pattern** column, which indicates the number of expressions of a condition that must occur before an incident is raised.
- The **Severity** column, which indicates the severity of the rule.
- The **Action** column, which indicates the action that is to be taken when incidents occur.
- The **Test Action** column, which allows you to test action settings by immediately triggering a condition's defined action.

For information about the Add Rule link, see the [Adding Rules](#) section.

For information about deleting rules, please see the [Deleting Rules](#) section.

## Adding Rules

To add a rule:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to add a rule.
2. On the **Rules** tab (**Performance Manager > Configuration > Rules**), click **Add Rule**.  
The **Configure Rule - Create Condition Expression** page appears. This page allows you to combine conditions and operators into rules.
3. Select a preconfigured condition from the **Condition** list. A description of the selected condition appears to the right of the list box.



**Note:** If the rule you are configuring involves multiple conditions, click **Add New Expression Part** to add a new **Condition** list and an **Operator** list. You can then use the up-and-down **Move** arrows to move the conditions within the expression. To delete a condition from the expression, click the corresponding **Delete** button.

4. Specify a frequency interval that will trigger an incident: **All the time**, **75% of times**, **50% of times**, **25% of times**, or **Once** within an interval of **15 minutes**, **30 minutes**, **45 minutes**, **1 hour**, **4 hours**, **6 hours**, **8 hours**, **12 hours**, **16 hours**, or **1 day**. Alternatively, use the **Raise incident if the expression is true <x> times in series** option to configure the rule to raise incidents after a certain number of consecutive failures, independent of any time frame.

The specified frequency interval begins as soon as you save the rule. Time intervals are used rather than numbers of executions because conditions may come from different transactions with differing schedules. To illustrate how condition frequency factors into rule expressions, consider a monitor schedule interval of 1 minute and a rule expression that raises an incident if the expression is true All the time within an interval of 15 minutes. In such an instance an incident will only be raised if the expression is true 15 times in a row-and the incident will not be raised until the 15 minute interval has concluded.

- If 50% of times is selected, an incident will be raised if the expression is true 8 of the 15 minutes.
- If Once is selected, an incident will be raised whenever the expression is true-regardless of whether the interval has concluded.

Condition frequency must also be considered when applying a rule to multiple monitors. Assume that we have a single rule with 2 conditions using OR logic and the frequency set to 2 times in a series and this rule applies to two monitors that run alternately. If the first monitor always fails and the second monitor always succeeds, the rule will never trigger because after every failure of monitor 1, monitor 2 reports a success, so the 2 times in a series frequency may not deliver the desired result for this scenario. A time-based frequency or multiple rules need to be used in this case.

5. Click **Next**. The **Configure Rule - Rule Name** page appears.
6. Enter a name for the rule in the **Rule Name** field.
7. Select an appropriate severity for the rule from the **Severity** list:
  - **Informational**: This is the lowest severity and is color-coded blue. This severity is used to mark incidents that do not indicate system problems. Although it can be used for rules, it is more likely to be used for custom incidents (for example, maintenance work or invalid results).
  - **Warning**: Color-coded yellow, this severity is used to define lower thresholds and offers early warning of potential problems. This is less likely to be used for custom incidents.
  - **Error**: Color-coded red, this severity is used for critical thresholds when immediate action is required. This severity is not likely to be linked to a log only action.
  - **Service Target Violation**: This severity is also color-coded red. This is a special type of error that is used as the basis for service target agreement statistics calculation. When a service target incident is active, a system is considered to be in a state where the service target of the service provider is in violation of agreed upon quality standards (for example, response times or availability). Service target statistics include:
    - **# Violations**: Number of times the service target is violated during the relevant interval
    - **Average duration**: Average duration of service target violations
    - **Downtime**: Percentage of time service target is violated during a time interval
    - **Uptime**: 100% minus Downtime%
    - **MTBF (mean time between failure)**: Average time from the end of one violation to the beginning of the next violation.

Example:

```
Time interval is 01:00 AM - 11:00 AM
Violation 1: 2:00 - 4:00 AM
Violation 2: 3:00 - 5:00 AM (based on a different rule)
Violation 3: 7:00 - 8:00 AM
```

This results in:

```
# Violations: 2 (the overlapping violations are counted as a single incident)
Avg. duration: 2 hours (3 hours and 1 hour divided by 2)
Uptime: 60% (4 hours of downtime in 10 hours)
MTBF: 3 hours (6 hours uptime / 2 incidents)
```

8. Specify the action that is to be taken when incidents occur. See [Configuring Rule Actions](#) for details.
9. Click **Next** to return to the **Rules** page.

# Editing Rules

To add a rule:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to edit a rule.
2. On the **Rules** tab (**Performance Manager > Configuration > Rules**), click the rule you want to edit. The **Configure Rule - Create Condition Expression** page appears. This page allows you to combine conditions and operators into rules.
3. Select a preconfigured condition from the **Condition** list. A description of the selected condition appears to the right of the list box.



**Note:** If the rule you are configuring involves multiple conditions, click **Add New Expression Part** to add a new **Condition** list and an **Operator** list. You can then use the up-and-down **Move** arrows to move the conditions within the expression. To delete a condition from the expression, click the corresponding **Delete** button.

4. Specify a frequency interval that will trigger an incident: **All the time**, **75% of times**, **50% of times**, **25% of times**, or **Once** within an interval of **15 minutes**, **30 minutes**, **45 minutes**, **1 hour**, **4 hours**, **6 hours**, **8 hours**, **12 hours**, **16 hours**, or **1 day**. Alternatively, use the **Raise incident if the expression is true <x> times in series** option to configure the rule to raise incidents after a certain number of consecutive failures, independent of any time frame.

The specified frequency interval begins as soon as you save the rule. Time intervals are used rather than numbers of executions because conditions may come from different transactions with differing schedules. To illustrate how condition frequency factors into rule expressions, consider a monitor schedule interval of 1 minute and a rule expression that raises an incident if the expression is true All the time within an interval of 15 minutes. In such an instance an incident will only be raised if the expression is true 15 times in a row-and the incident will not be raised until the 15 minute interval has concluded.

- If 50% of times is selected, an incident will be raised if the expression is true 8 of the 15 minutes.
- If Once is selected, an incident will be raised whenever the expression is true-regardless of whether the interval has concluded.

Condition frequency must also be considered when applying a rule to multiple monitors. Assume that we have a single rule with 2 conditions using OR logic and the frequency set to 2 times in a series and this rule applies to two monitors that run alternately. If the first monitor always fails and the second monitor always succeeds, the rule will never trigger because after every failure of monitor 1, monitor 2 reports a success, so the 2 times in a series frequency may not deliver the desired result for this scenario. A time-based frequency or multiple rules need to be used in this case.

5. Click **Next**. The **Configure Rule - Rule Name** page appears.
6. Enter a name for the rule in the **Rule Name** field.
7. Select an appropriate severity for the rule from the **Severity** list:
  - **Informational:** This is the lowest severity and is color-coded blue. This severity is used to mark incidents that do not indicate system problems. Although it can be used for rules, it is more likely to be used for custom incidents (for example, maintenance work or invalid results).
  - **Warning:** Color-coded yellow, this severity is used to define lower thresholds and offers early warning of potential problems. This is less likely to be used for custom incidents.
  - **Error:** Color-coded red, this severity is used for critical thresholds when immediate action is required. This severity is not likely to be linked to a log only action.
  - **Service Target Violation:** This severity is also color-coded red. This is a special type of error that is used as the basis for service target agreement statistics calculation. When a service target incident is active, a system is considered to be in a state where the service target of the service provider is in

violation of agreed upon quality standards (for example, response times or availability). Service target statistics include:

- **# Violations:** Number of times the service target is violated during the relevant interval
- **Average duration:** Average duration of service target violations
- **Downtime:** Percentage of time service target is violated during a time interval
- **Uptime:** 100% minus Downtime%
- **MTBF (mean time between failure):** Average time from the end of one violation to the beginning of the next violation.

Example:

```
Time interval is 01:00 AM - 11:00 AM
Violation 1: 2:00 - 4:00 AM
Violation 2: 3:00 - 5:00 AM (based on a different rule)
Violation 3: 7:00 - 8:00 AM
```

This results in:

```
# Violations: 2 (the overlapping violations are counted as a single incident)
Avg. duration: 2 hours (3 hours and 1 hour divided by 2)
Uptime: 60% (4 hours of downtime in 10 hours)
MTBF: 3 hours (6 hours uptime / 2 incidents)
```

8. Specify the action that is to be taken when incidents occur. See [Configuring Rule Actions](#) for details.
9. Click **Next** to return to the **Rules** page.

## Deleting Rules

To delete a rule:

1. On the **Projects** page (**Performance Manager > Projects**), click the project from which you want to delete a rule.
2. On the **Rules** tab (**Performance Manager > Configuration > Rules**), click the **Delete** button that corresponds to the rule you want to delete.  
The **Delete Rule** confirmation message appears.
3. Click **Delete** to permanently delete the rule.

## Configuring Rule Actions

Actions determine what Performance Manager does when incidents occur. See [Rules](#) for details regarding rule expressions. To configure a rule action:

1. Enter your first step here. Enter the result of your step here (optional).
2. While adding or editing a rule expression (see [Editing Rules](#) for details), specify an action to be taken on the **Configure Rule - Rule Name** page.
3. Complete the subsequent action profile page based on the settings of the action you selected.

Here is a list of available actions, along with brief descriptions and instructions for completing the corresponding action profile pages:

- **Log-only:** Select this action to simply have incidents logged on the **Incidents** report (**Performance Manager > Reports**).
- **Email notification:** Sends email notifications to specified users.
  1. Define who is to receive notifications using the **To**, **CC**, and **BCC** fields. Multiple email addresses can be entered (separated by commas).

2. Enter the name that is to appear in notifications' **From** fields in the page's **From** field. It is recommended that this field contain the email address of the rule creator.
  3. Enter the email address that email responses are to be directed to in the **Reply-To** field. This is the person who should be informed of rule violations.
  4. Enter a notification message in the **Message text** field.
  5. Click **Save**.
- **Execute Action Essential:** Action Essentials are Essentials that are used as powerful notification alarms. When an action Essential provides additional attributes, you are prompted here with a project attributes dialog; edit the attributes of the action Essential as required. Action Essentials must be preconfigured by system administrators. See the "Managing Essentials" section of the Performance Manager Administration Help for details regarding action Essentials.
  - **Pager-Message:** Sends a pager message to a specified user.
    1. Define who is to receive the pager message using the **Registered Recipient Name** field.
    2. Enter a notification message in the **Custom text message** field.
    3. Click **Save**.
  - **SMS-Message:** Sends an SMS message to a specified user.
    1. Define who is to receive the pager message by entering a telephone number in the **Telephone number** field.
    2. Enter a notification message in the **Custom message text** field.
    3. Click **Save**.
  - **SNMP-Trap:** Sends an SNMP-Trap message to a specified host that has an installed SNMP daemon that receives SNMP traps. SNMP traps can be used to expose information to Enterprise Management Systems (EMS).
    1. Define who is to receive the message by entering a host name in the **SNMP Host** field.
    2. Enter a port number in the **SNMP Port** field.
    3. Click **Save**.
  - **Change Schedule:** Changes the interval of a monitor in case of an incident. The schedule is set back to its original interval as soon as the incident closes.



**Note:** If multiple rules try to change a schedule simultaneously, only the first rule is activated and changes the schedule.

- Select the **Monitors** for which the interval changes in case of an incident. All monitors are listed for which the selected rule has a depending condition.
- Specify the **Interval** (day, hour, minute) at which the monitor should execute while the incident is active.

## Custom Incidents

Incidents are comparable to error conditions. Incidents are raised when predefined conditions are met (for example, when performance drops below a certain minimum threshold for a certain length of time and a service target violation is triggered).

Custom incidents are different in that they are not incidents that are detected by Performance Manager. They are incidents that are detected by users and manually logged into Performance Manager. Custom incidents that override system-detected incidents can be defined. Custom incidents are typically used to

manually override incidents such as service target violations with custom incidents that have the **Severity** set to **Informational**, which resets service target violations; or the severity setting **Service Target Violation**, which sets a service target violation. Custom incidents are implemented project wide and cannot be assigned to specific rules.

Custom incidents are logged on the **Custom Incidents** page. (**Performance Manager > Configuration > Custom Incidents**).

### The Custom Incidents Page

The **Custom Incidents** page lists all custom incidents that occur during the time period selected using the calendar tool.

- The beginning-time and end-time of each custom incident is listed in the **From** and **To** columns.
- The duration of each custom incident is listed in the **Duration** column.
- The error type (**Service Target Violation**, **Error**, **Warning** or **Informational**) is listed in the **Type** column.

Custom incidents can be sorted by column (**Name**, **From**, **To**, **Duration**, and **Type**) by clicking the column names or the **Ascending/Descending** arrow links. To change sort-order-by-column from ascending to descending (and vice-versa), click the appropriate **Ascending/Descending** arrow links.

To view detailed information of a specific custom incident, click the **Report** icon of the incident you want to view in **Monitoring > Incidents**. The **Incident report** also enables you to jump directly to the relevant time slot within the health report when the incident occurred.

## Adding Custom Incidents

To add a custom incident:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to add a custom incident.
2. On the **Custom Incidents** tab (**Performance Manager > Configuration > Custom Incidents**), click **Add Custom Incident**.  
The **Add Custom Incident** page appears.
3. Enter a name for the custom incident in the **Incident Name** field.
4. Enter a description of the custom incident in the **Description** field.
5. Specify start and end times for the incident using the **From** and **To** lists.
6. Specify whether the custom incident should override system-detected health monitoring results for the same time period by checking the **Override System incidents** check box.
7. Select an appropriate severity for the custom incident from the **Severity** list.
  - **Informational**: This is the lowest severity and is color-coded blue. This severity is used to mark incidents that do not indicate system problems. Although it can be used for rules, it is more likely to be used for custom incidents (for example, maintenance work or invalid results).
  - **Warning**: Color-coded yellow, this severity is used to define lower thresholds and offers early warning of potential problems. This is less likely to be used for custom incidents.
  - **Error**: Color-coded red, this severity is used for critical thresholds when immediate action is required. This severity is not likely to be linked to a `log only` action.
  - **Service Target Violation**: This severity is also color-coded red. This is a special type of error that is used as the basis for service target agreement statistics calculation. When a service target incident is active, a system is considered to be in a state where the service target of the service provider is in violation of agreed upon quality standards (for example, response times or availability). Service target statistics include:
    - **# Violations**: Number of times the service target is violated during the relevant interval

- **Average duration:** Average duration of service target violations
- **Downtime:** Percentage of time service target is violated during a time interval
- **Uptime:** 100% minus Downtime%
- **MTBF (mean time between failure):** Average time from the end of one violation to the beginning of the next violation.

8. Click **Save** to save the custom incident.

## Editing Custom Incidents

To edit a custom incident:

1. On the **Projects** page (**Performance Manager > Projects**), click the project for which you want to edit a custom incident.
2. On the **Custom Incidents** tab (**Performance Manager > Configuration > Custom Incidents**), click the name of the custom incident you want to edit .  
The **Edit Custom Incident** page appears.
3. Enter a name for the custom incident in the **Incident Name** field.
4. Enter a description of the custom incident in the **Description** field.
5. Specify start and end times for the incident using the **From** and **To** lists.
6. Specify whether the custom incident should override system-detected health monitoring results for the same time period by checking the **Override System incidents** check box.
7. Select an appropriate severity for the custom incident from the **Severity** list.
  - **Informational:** This is the lowest severity and is color-coded blue. This severity is used to mark incidents that do not indicate system problems. Although it can be used for rules, it is more likely to be used for custom incidents (for example, `maintenance work` or `invalid results`).
  - **Warning:** Color-coded yellow, this severity is used to define lower thresholds and offers early warning of potential problems. This is less likely to be used for custom incidents.
  - **Error:** Color-coded red, this severity is used for critical thresholds when immediate action is required. This severity is not likely to be linked to a `log only` action.
  - **Service Target Violation:** This severity is also color-coded red. This is a special type of error that is used as the basis for service target agreement statistics calculation. When a service target incident is active, a system is considered to be in a state where the service target of the service provider is in violation of agreed upon quality standards (for example, response times or availability). Service target statistics include:
    - **# Violations:** Number of times the service target is violated during the relevant interval
    - **Average duration:** Average duration of service target violations
    - **Downtime:** Percentage of time service target is violated during a time interval
    - **Uptime:** 100% minus Downtime%
    - **MTBF (mean time between failure):** Average time from the end of one violation to the beginning of the next violation.
8. Click **Save** to save the custom incident.

## Deleting Custom Incidents

To delete a custom incident:

1. On the **Projects** page (**Performance Manager > Projects**), click the project from which you want to delete a custom incident.
2. On the **Custom Incidents** tab (**Performance Manager > Configuration > Custom Incidents**), click the **Delete** button that corresponds to the custom incident you want to delete.  
The **Delete Custom Incident** confirmation message appears.

3. Click **Delete** to permanently delete the custom incident.

# Analyzing Results

This section explains how to interpret Performance Manager monitoring results.

## Overview

Health rate calculation offers a single measurement that reflects overall system health-enabling analysts to detect high-level problems at a glance, even if they have little familiarity with the system under test. By reversing health rate calculations, analysts can readily see whether an error is an availability problem, an accuracy problem, or a performance problem. Ultimately, analysts can drill down to the relevant components that generate errors.

Health rates don't replace data measurements; they simply contextualize them. Specific measurement values remain important for in-depth analysis, experienced analysts, service target agreements, and notifications.

### Benefits of health rates

The benefit of health rates lies in the fact that they offer analysts a short cut in evaluating project health and directing development efforts. If the overall health rate of a project is high, there is no need for further analysis. Health rates have values between "0" and "100" ("0" being the worst, "100" being the best) and are independent of projects, amounts of data analyzed, and frequency of individual measurements.

Because high-level health rates are the aggregate of low-level health rates, analysts have the option of reversing rate calculations to determine how specific low-level rates influence overall rates. Such causal analysis can be used to "drill-down" to specific low-level data that is negatively affecting overall rates—thereby pinpointing the system components that are having a negative impact on system health. All the while, measurements that fall within acceptable ranges can be disregarded.

Because low-level health rates reveal the fitness of actual measurement values, analysts typically don't need to understand the significance of the measurement values themselves. For example, without having familiarity with a certain monitored application, it isn't readily apparent whether a business transaction that takes 15 seconds is faster or slower than usual. A health rate of "95%" however is readily understood to be a healthy rate.

### Calculating health rates

Performance Manager is designed to monitor your perception of your system's health. You specify boundaries (100% and 0%) between which performance health is calculated as a logarithmic function. Outside of these boundaries health is considered to be either 100% or 0%. In scenarios where baseline information from which meaningful boundaries could be derived isn't available, Performance Manager can configure boundaries based on historic traffic patterns.

See [Calculating Health](#) for more details regarding health-rate calculation.

### Health dimensions

Overall health values are influenced by three health dimensions:

- Availability
- Accuracy
- Performance

Each of these health dimensions and the overall health rate itself provide values in the range of "0" to "100"—the higher the value, the better the health of the system.

Only availability and accuracy health values are expressed as percentages. Performance and overall health values are expressed as absolute rates ("10," "20," "30," etc.).

When you review health dimension values offered by Performance Manager, you are normally reviewing values that were calculated based on multiple monitoring transactions. So, when evaluating health dimension values, keep in mind that the health dimension value for a set of monitoring transactions is equal to the average value of all the corresponding and existing health dimension values of all individual monitoring transactions.

### **Availability**

Availability is the most basic health dimension. It measures the percentage of time during which a monitored system is available to a subset of selected data. The availability rate provides information regarding whether a monitored system is running and whether it provides basic responsiveness to client requests.

A system is judged "available" when a monitoring transaction testing a system completes without detecting any errors. Most errors indicate that a system is not available. Exceptions include those errors that indicate that a system is available, but not working correctly.

When several monitors supervise a system and some of those monitors detect that the system is not available while other monitors detect that the system is available, the availability of the system is rated in between 0% and 100%.

### **Accuracy**

The Accuracy rating for monitored systems is calculated only after systems are judged "available."

Accuracy rates are calculated with the assumption that monitored systems are working as designed and that the information transmitted to clients is correct. Useful functions that can be evaluated to determine accuracy include link checking, content validation, title validation and response data verification. If a monitoring script contains customized functions that are used to ascertain system accuracy, those functions will be factored into the accuracy rate as well.

Accuracy rating goes far beyond the simple checking of availability. A server may be available even when the application it hosts isn't responding. Likewise, dynamic pages may be corrupt, database queries may produce empty result sets, and warehouses may run short of stocked merchandise. The simple checking of availability won't alert one to such failures. Only complex transactions that compare results to benchmarks will detect these problems.

### **Performance**

Once a monitored system is judged available and accurate, the performance health dimension of a system is calculated. Performance is not as objective a measure as availability and accuracy; what qualifies as "good" and "bad" performance is subjective, varying from one system to the next.

The most common measures of performance are timers. Users who are quite familiar with the behavior of their systems can have performance rates calculated against established timer results-determined through baseline testing.

For users who have not run baseline tests with Silk Performer to determine baseline performance-and consequently do not know what the boundaries for good and bad performance are-Performance Manager offers a means of calculating performance ratings based on historical response time values. By calculating dynamic bounds for "good" and "bad" performance based on historical data, actual performance values of monitor executions can be compared against historic values of monitor executions to determine system performance.

# Analyzing Client and Infrastructure Health

Client and infrastructure health reports offer a tremendous amount of monitoring information and can be manipulated in a number of ways to isolate relevant monitoring data.

Average, minimum, and maximum ratings for each health component are listed in the **Avg**, **Min**, and **Max** columns respectively.

**Health**, **Availability**, **Accuracy**, or **Performance** can be selected to restrict the data shown in a diagram to a specific dimension. Only selected dimensions influence heat fields and are reflected in report details.

## Selecting a range from the calendar

The **Select range** feature and the calendar feature are available throughout Performance Manager's **Monitoring** section (**Performance Manager > Monitoring**). The calendar's **From** and **To** rows allow you to specify start and end times for the period of time for which you want to view health statistics.

After specifying **From** and **To** times with the lists, click **Update** to update the report based on the new time range.

The **hour**, **day**, **week**, **month**, [**last 24 hours**] links allow you to bypass the drop-down calendar and simply view statistics for the most recent time period (past hour, past day, past week, past month, or past 24 hours).

You may also use the **Forward** and **Backward** arrows to increase/decrease the selected time range by specified intervals (one hour, one day, one week, or one month).

The magnifying glass **Increase** and **Decrease** links are useful for enlarging and reducing the range of time covered by the report. The **Increase** link enlarges the period of time by 50%. The **Decrease** link reduces the period of time by 50%.

After specifying a new time period, click **Update** to update the report.

If the calendar displays a custom interval, for example after zooming in or out, you can use the left-most arrows ( **Earlier/Later**) to increase/decrease the selected time range by half of the selected interval.

## Transactions

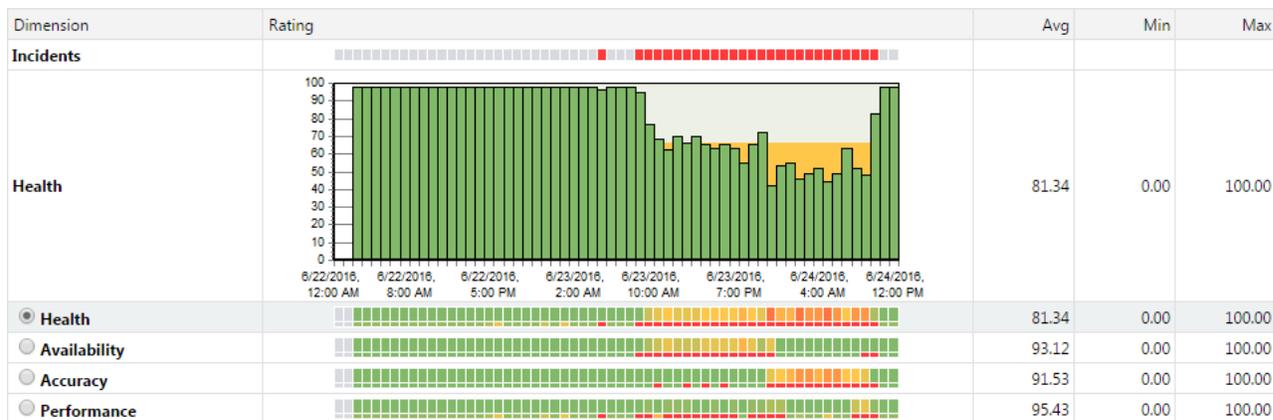
The **Transaction** portion of the client and infrastructure health reports reveals overall health statistics for each project transaction. Note that the time intervals in the **Transaction** section correlate with the time intervals of all other sections of the health report. This allows you to isolate detected problems by drilling down to more granular data. For example, an overall poor health rating might be the result of one of three transactions experiencing an error while the other two transactions are performing well. Isolating which transaction an error condition is tied to may speed up defect turnaround time.

Average, minimum, and maximum overall health ratings for each transaction are listed in the **Avg**, **Min**, and **Max** columns respectively.

## Locations

(Client health only): The **Location** section of the client health report lists all of the locations where Performance Manager agents run monitors. It may be that the monitors at one location are returning availability errors while monitors at other locations are not experiencing problems. The **Location** section of the report might indicate that this is the result of network problems at the one troubled location-and not the result of global application unavailability.

Average, minimum, and maximum ratings for each location are listed in the **Avg**, **Min**, and **Max** columns respectively.



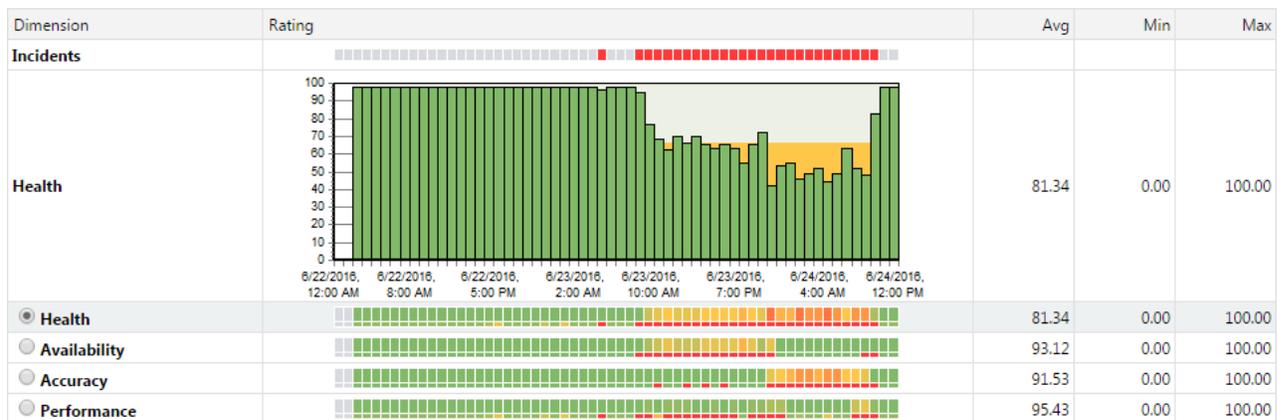
## Interpreting Health Detail Reports

Performance Manager heat fields offer a quick shorthand for conveying health rates.

Each field represents a time interval and each field's color indicates the health during that time interval:

- Heat fields are used for each of the three health dimensions (performance, accuracy, and availability) in addition to overall health.
- Health dimensions use a range of 0 to 100.
- Green (66 - 100)
- Yellow (33 - < 66)
- Red ( 0 - < 33)
- Performance is less than 100, but greater than 80 when performance values for the monitor are less than bound1 (the bound that indicates "good" performance).
- Performance equals 0 when performance values for the monitor are greater than bound2 (the bound that indicates "bad" performance).
- Performance is greater than 50 when performance values for the monitor are less than bound2.
- All performance values between bound1 and bound2 are between 100 and 50 and are interpolated using a logarithmic function.

See [Interpreting Measured Data](#) for more information regarding boundaries.



Each time interval, or column, is broken into six rows—each row represents a different dimension of system health:

- **Incidents** - Red fields indicate errors or service target violations. Blue fields are informational. Yellow fields indicate warnings. No differentiation is made between custom incidents and system incidents.
- **Health** (detailed) - Fields in this row are not heat fields. This row is a histogram that displays overall health as a value between 0 and 100. Health values between 33 - 66 are indicated by yellow. Health values of 0 to 32 are indicated by red.
- **Health** (summary) - These heat fields signify aggregate health, factoring in system availability, accuracy, and performance.
- **Availability** - These health fields signify system availability.
- **Accuracy** - These health fields signify system accuracy.
- **Performance** - These health fields signify system performance.

Each heat field is comprised of two elements: (1) a larger box that signifies the average health value recorded during the interval, and (2) beneath that, a smaller box, which signifies the minimum health value recorded during the interval.

To view a small "roll over" dialog that details a specific time interval, hold your cursor over a time interval of interest.

To progressively zoom into chart details (and thereby limit the amount of time covered by a chart), click a time interval of interest. To zoom out of a chart, use the Less ("-") magnifying glass link of the calendar tool at the top.

## Project Overview Reports

The Performance Manager **Project Overview Report** accommodates the needs of both technical and business-management users by offering different layouts of the project overview report.

### Project Overview Report Page

To address the reporting requirements of business-management users, Performance Manager offers reports that provide a bird's-eye view of the project status and allow for "guided drill-downs" that don't require deep technical knowledge. Such reports support decision-makers by offering holistic insight into system status and customer satisfaction.

#### Health Status

The **Health Status** tab (**Performance Manager > Projects > Health Status**) offers a **Trend Overview Report** that informs business managers of the status of projects - steady, improving, or decreasing.

The health status of each project is indicated by the color (green, yellow, or red) of the square button to the left of each project name; and also as a numerical value in the Health column.

The direction and color of trend arrows in the Trend columns indicate the project status. Green arrows indicate that the project status is improving, white arrows indicate that the status is steady, and red arrows indicate that the status is decreasing.

You can configure parameters for one or two trend columns on this page. Trends can be defined to tell you the status of system health during the last 7 days of the past month or the last 24 hours of the past week. The Health column can be configured to reflect the past 24 hours, the past 7 days, or the past 31 days.

### Health Drilldown

The **Health Drilldown** tab (**Performance Manager > Projects > Health Drilldown**) offers all of the features included on the **Health Status** tab, but enables detailed analysis of health status and trends through drilldown functionality. This information is tailored to more technical users (for example, head of development). Details offered here enable you to determine if detected problems originate with the infrastructure, the client, or the application. From there you can drill down through transactions to specific health dimensions (availability, performance, and accuracy).

### Health History

The **Health History** tab (**Performance Manager > Projects > Health History**) is useful for comparing current project health with historical developments in project health. This is visualized by a heat field that indicates the health rating for the currently selected time period. The current health rating is shown as a heat field item on the left. At a glance, this report allows you to determine if current system status falls within historical limits or if it significantly deviates from historical limits-giving you an indication of the relevance of problems.

### Service Target Status

The **Service Target Status** tab (**Performance Manager > Projects > Service Target Status**) is ideally suited for business line managers who are interested in service target performance ratings, as defined by fulfillment ratings that have been established for projects. The trends detailed on this tab do not apply to project health but rather to the development of service target fulfillment ratings. Thresholds are dictated by overall service target ratings.

For each project, **Service Target Status** can be calculated with one or two trends over independent time frames.

### Snapshot

Although it shares some of the same functionality as the **Health Drilldown** report, the **Snapshot** tab (**Performance Manager > Projects > Snapshot**) offers a snapshot of only the most recent result set. This information is ideally suited for IT departments who only need to determine whether a system is up and running accurately, not what past trends were. The last monitor execution measurement is recorded with a timestamp.

The **Root Cause** feature is also available here, enabling ad hoc diagnostics by pinpointing the sources of detected errors.

## Customizing Your Start Page

Specify which reports are included in your overview report, what time frames the reports and trends cover, and which report appears as your start page when you log into Performance Manager.

Both administrators and users can customize these settings to meet their individual needs. To customize your start page:

1. Go to **Performance Manager > Projects > Overview**.

2. Click **Customize Project Lists**.
3. Select the **Include inactive projects in Health Status, Health Drilldown, and Health History View** check box to have inactive projects displayed alongside active projects.
4. Select the **Use calendars in Health History and Service Target Status View instead of static timespans** check box to have Performance Manager's calendar feature appear on all report pages.



**Note:** This setting overrides the time frame settings you define in the **Health Status** and **Service Target Status** sections, instead offering calendars to query your desired time frames.

5. In the **Health Status** section, select the **Start page** check box to have the Health Status report appear as your start page. Select the **Show tab** check box to have this report appear in your overview report, but not be your start page.
6. Specify the time period that should be covered on the Health Status report by selecting a time frame from the drop-down list, for example **last 24 hours**, **last 7 days**, or **last 31 days**.
7. In the **Health Status - Trend 1** row, select the time frame you want for the first trend, for example **last 24 hours vs. last 7 days** or **last 7 days vs. last 31 days**. Select **None** to disable this trend.
8. In the **Health Status - Trend 2** row, select the time frame you want for the second trend, for example **last 24 hours vs. last 7 days** or **last 7 days vs. last 31 days**. Select **None** to disable this trend.
9. In the **Health Drilldown** section, select the **Start page** check box to have the Health Drilldown report appear as your start page. Select the **Show tab** check box to have this report appear in your overview report, but not be your start page. The **Health Drilldown** section uses the time period settings of the **Health Status** view.
10. In the **Health History** section, select the **Start page** check box to have the Health History report appear as your start page. Select the **Show tab** check box to have this report appear in your overview report, but not be your start page.
11. Specify the time period that should be covered on the Health History report by selecting a time frame from the list, for example **last 24 hours**, **last 7 days**, or **last 31 days**.
12. In the **Service Target Status** section, select the **Start page** check box to have the Service Target Status report appear as your start page. Select the **Show tab** check box to have this report appear in your overview report, but not be your start page.
13. Specify the time period that should be covered on the Service Target Status report by selecting a time frame from the list, for example **last 24 hours**, **last 7 days**, or **last 31 days**.
14. In the **Service Target Status - Trend 1** row, select the time frame you want for the first trend, for example **last 24 hours vs. last 7 days** or **last 7 days vs. last 31 days**. Select **None** to disable this trend.
15. In the **Service Target Status - Trend 2** row, select the time frame you want for the second trend, for example **last 24 hours vs. last 7 days** or **last 7 days vs. last 31 days**. Select **None** to disable this trend.
16. In the **Snapshot** section, select the **Start page** check box to have the Snapshot report appear as your start page. Select the **Show tab** check box to have this report appear in your overview report, but not be your start page.
17. Click **Save** to save your start page customization.

## Emailing Reports

Performance Manager enables you to email project overview reports directly from its interface without opening your mail client.

To email a project overview report:

1. Click the **Email this Report** link on any project overview report tab (**Performance Manager > Projects**). The **Configure email settings for sending this report** page appears, pre-configured with your email user settings (**Administration > Users**).
2. Specify the recipient of the report in the **To** line.

3. Add a secondary **CC** recipient if necessary.
4. Specify a **Reply-to** email address.
5. Specify a title for your email in the **Subject** field.
6. Click **Send** to send the report.

## Contacting Project Owners

Performance Manager enables you to send emails to project owners via the **Contact Project Owner** links on each project overview page.

To send an email to a project owner:

1. From any project overview report tab other than **Overview (Performance Manager > Projects)**, click a project owner's name in the **Contact Project Owner** column. Your email client opens.
2. Send the email as specified by your email client.

## Interpreting Measured Data

Interpret your monitoring results by analyzing the measured availability, accuracy and performance data.

### Availability

The **Availability** portion of the **Project Overview** report (**Performance Manager > Monitoring > Client Health**) offers details regarding specific availability errors that were received during the selected time range. Example errors include `Connection timed out`, `Connection refused`, and `No route to host`.

To display the detailed information described below, click **Show**.

#### # Availability Errors

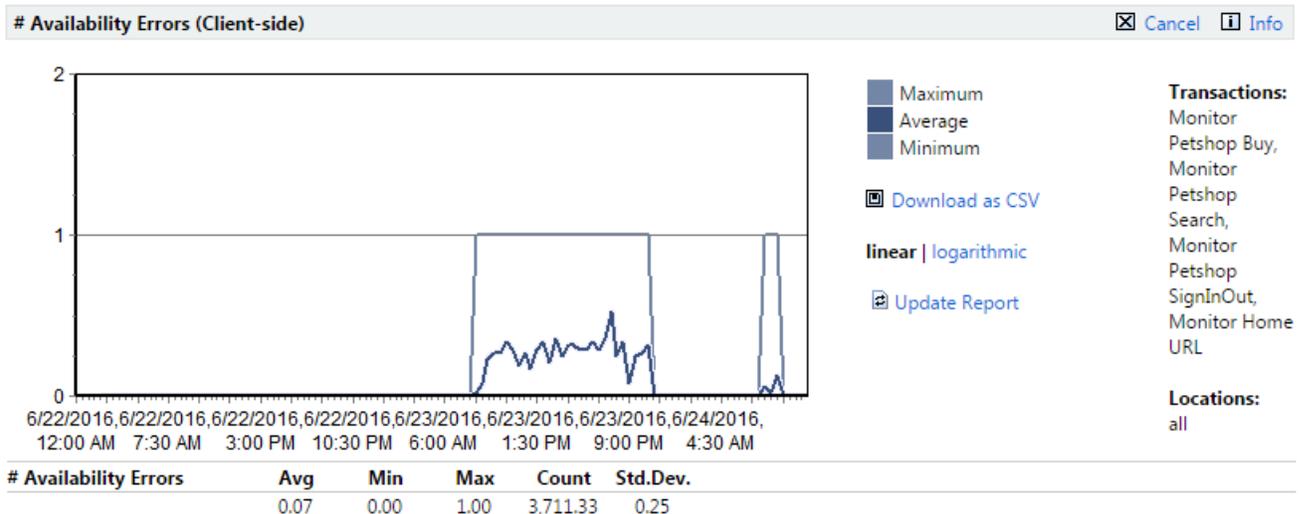
Shows the number of availability errors within the selected time span.

This section offers the base values:

- Average (Avg)
- Minimum (Min)
- Maximum (Max)
- Standard deviation (Std. Dev): An indication as to the stability of a measurement. If the number of errors is more or less always the same, the **Std. Dev** value will be relatively small. A large **Std. Dev** value indicates that the number of errors is variable.

#### Availability Error Count Detail Chart

To see detailed analysis of the number of availability errors over time, click **# Availability Errors**. This takes you to the availability **Details** chart.



The chart shows how many availability errors occurred at a specific time. The base values are listed as well.

### Availability Messages

The **Count** column indicates how often this value was measured during a given time frame, independent of whether an error was actually reported or not.

Click an availability error to view a detailed listing of each of the transactions that resulted in that particular availability error.

Each error listing includes **Timestamp**, **Message**, **Location**, and **Transaction** details.

Errors can be sorted by clicking the **timestamp** and **message** column names or the **Ascending/Descending** arrow links. To change sort-order-by-column from ascending to descending (and vice-versa) click the appropriate **Ascending/Descending** arrow link.

Error listings offer the option of downloading `.wrt` files and `.xlz` TrueLog files. `.wrt` files contain information that is written with `write` and `writeln` Silk Performer script commands.

TrueLog information is tracked to support root-cause analysis of reported errors. TrueLog files can be downloaded as WinZip archives (extension `.zip`). "Zipped" TrueLog files are automatically extracted when opened with TrueLog Explorer and are recommended for use when limited bandwidth is an issue. You can click any monitor execution to display detailed diagrams that cover monitor-execution statistics such as page times, connection times, and handshake times. See the [Working with Silk Performer](#) section of the Introduction for details regarding TrueLog technology.

TrueLog files are only available when they are enabled in monitor settings. See [Adding Monitors](#) for details regarding enabling the writing of TrueLog files. Available TrueLog files are indicated with **Download Files** buttons. See [Downloading TrueLog Files](#) for details.

## Accuracy

The **Accuracy** section of the **Project Overview** report (**Performance Manager > Monitoring > Client Health**) offers details regarding accuracy errors that were received during the selected time range. Example errors include `HTML form not found`, `Unexpected connection close during read`, and `HTML hyperlink not found`.

### # Accuracy Errors

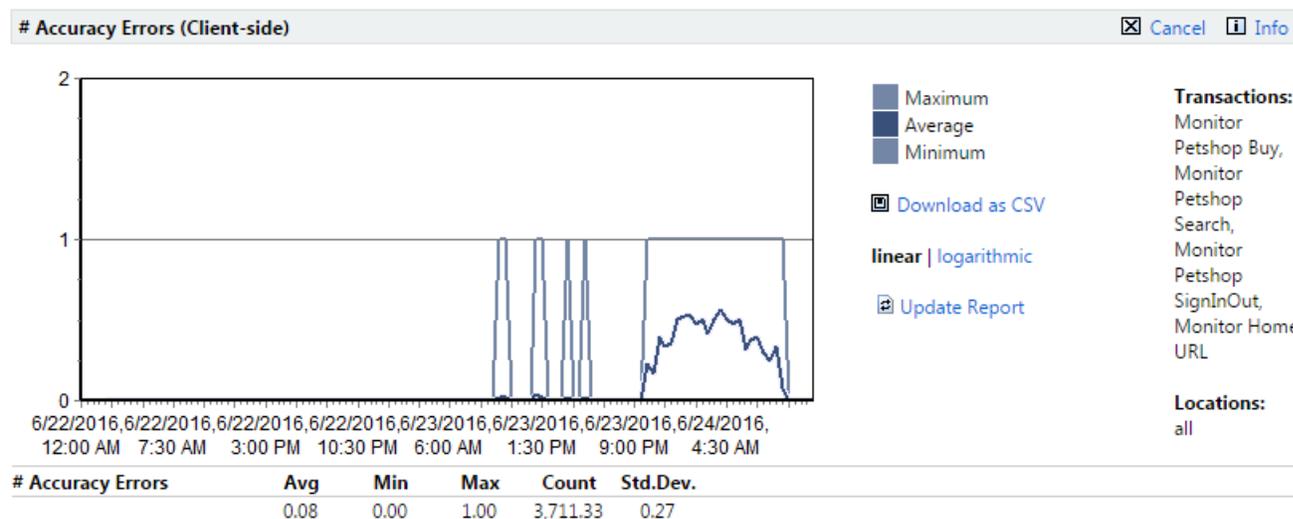
Shows the number of accuracy errors within the selected time span.

This section offers the base values:

- Average (Avg)
- Minimum (Min)
- Maximum (Max)
- Standard deviation (Std. Dev): An indication as to the stability of a measurement. If the number of errors is more or less always the same, the **Std. Dev** value will be relatively small. A large **Std. Dev** value indicates that the number of errors is variable.

### Accuracy Error Count Detail Chart

To see detailed analysis of the number of accuracy errors over time, click **# Accuracy Errors**. This takes you to the accuracy **Details** chart.



The chart shows how many accuracy errors occurred at a specific time. The base values are listed as well.

### Accuracy Messages

The **Count** column indicates how often this value was measured during a given time frame, independent of whether an error was actually reported or not.

Click an accuracy error to view a detailed listing of each of the transactions that resulted in that particular accuracy error.

Each error listing includes **Timestamp**, **Message**, **Location**, and **Transaction** details.

Errors can be sorted by clicking the **timestamp** and **message** column names or the **Ascending/Descending** arrow links. To change sort-order-by-column from ascending to descending (and vice-versa) click the appropriate **Ascending/Descending** arrow link.

Error listings offer the option of downloading **.wrt** files and **.xlsx** TrueLog files. **.wrt** files contain information that is written with `write` and `writeln` Silk Performer script commands.

TrueLog information is tracked to support root-cause analysis of reported errors. TrueLog files can be downloaded as WinZip archives (extension **.zip**). "Zipped" TrueLog files are automatically extracted when opened with TrueLog Explorer and are recommended for use when limited bandwidth is an issue. You can click any monitor execution to display detailed diagrams that cover monitor-execution statistics such as page times, connection times, and handshake times. See the [Working with Silk Performer](#) section of the Introduction for details regarding TrueLog technology.

TrueLog files are only available when they are enabled in monitor settings. See [Adding Monitors](#) for details regarding enabling the writing of TrueLog files. Available TrueLog files are indicated with **Download Files** buttons. See [Downloading TrueLog Files](#) for details.

## Performance

The **Performance** section of the project overview report is divided into the following areas:

- Transaction response time
- Page times
- Custom timers
- Custom counters

Each section offers the same base values:

- **Performance**
- Average (**Avg [s]**)
- Minimum (**Min [s]**)
- Maximum (**Max [s]**)
- Standard deviation (**Std. Dev**): An indication as to the stability of a measurement. If values are more or less always the same, the Std. Dev value will be relatively small. A large Std. Dev value indicates that a measure is quite variable.
- **Histogram** (see [Histograms](#))
- **Transaction response time**: Shows the average transaction response times for all selected transactions (see [Setting up Monitors](#)).
- **Page times**: Total page download time. To view detailed page statistics, go to the page detail report by clicking the page name.
- **Custom Timers**: Custom timers are created by the script functions `MeasureStart()` and `MeasureStop()` and can be used to measure individual timings of interest.
- **Custom Counters**: Custom counters are created by the script function `MeasureIncFloat` and can be used to measure individual components of interest (for example, custom counters are used in resource monitors to measure such things as CPU usage, memory usage, and throughput related measures).

The **Count** column indicates how often a timer or counter was measured during a given time frame.

### Boundaries

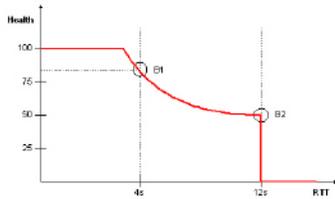
Performance Manager's Boundary Editor is the recommended approach for editing bounds in Silk Performer scripts. See [Configuring Static Boundaries for Performance Values](#) for details.

Performance rates are calculated by comparing measured values to boundary values. Outside of specified boundaries, performance is considered to be either "good" or "bad." Boundaries can be defined explicitly using Silk Performer's `MeasureSetBound` function or automatically by Performance Manager. See [Adding Monitors](#) for details.

If you defined boundaries explicitly, using Silk Performer's `MeasureSetBound` function, you should select **Performance Rate Calculation based on bounds defined in the script** when creating the monitor.

If you use automatically adjusted boundaries, or if you changed the boundaries in the script, the actual values of the boundary is displayed in parenthesis, the percentage indicates how many measurements were below or above the boundary whatever it was in the past. You can see the changes of the boundaries in the detail chart of the measurement.

Rather than using a percentile algorithm to establish performance health, raw data is assessed against two specific boundaries that allow performance to be established using a metric that is common to availability and accuracy ratings.



Based on the health function shown above, the first boundary (B1) is considered to be the ideal value. Deviations from this ideal are not critical, but they are reflected in health values. This is particularly important when there is a pattern of performance degradation. The emphasis of these deviations is supported by the exponential behavior of the function. The second boundary (B2) marks severe behavior, and so the health rating drops to zero.

The ability to define boundaries allows users to implement custom rating systems that are meaningful. Boundary values are typically obtained from the baselines of previous load tests. Examples include "70.00% < 8.00" (boundary 1) and "100.00% < 16.00" (boundary 2). Such boundaries indicate that 70% of the transactions are completed in under 8 seconds and 100% of the transactions are completed in under 16 seconds.

## Histograms

Histograms are heat field graphs that represent boundary-based values with colors ranging from green to red. This method of visualization allows for performance evaluation at a glance.

- The left sides of histograms indicate the percentage of measures below both bounds (good performance) in green.
- The middle indicates the percentage of measures higher than bound 1 and below bound 2 (performance warning level) in yellow.
- The right side indicates the percentage of measures above both bounds (poor performance) in red.

## Performance Detail Charts

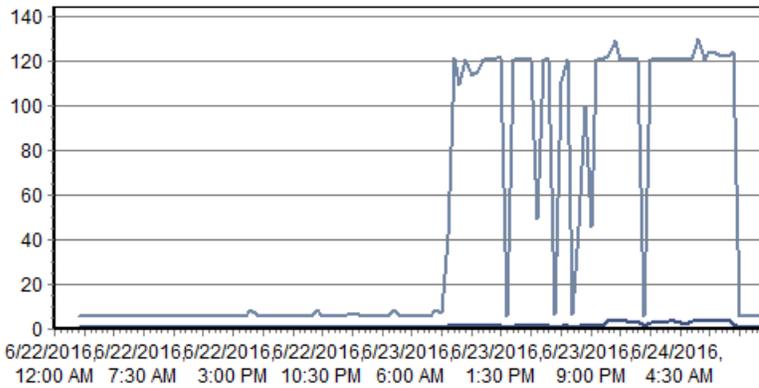
In addition to offering the same transaction-specific information that is available on the project overview report, performance details charts offer visual depictions of performance over time.

To view the response time of a specific transaction, you must de-select all other transactions in the **Transactions** section of the project overview report.



**Note:** When multiple transactions that use different boundaries are selected, boundaries are not displayed in the chart.

For example, to see a detailed analysis of the overall page time, click **Average Page Time** (under the **Page Times** node) . This takes you to a performance **Details** chart.



Maximum [s]  
Average [s]  
Minimum [s]  
No boundaries defined or merged boundaries differ.

- Download as CSV
- linear | logarithmic
- ▶ Show Page Details
- Update Report

**Transactions:**  
Monitor  
Petshop Buy,  
Monitor  
Petshop  
Search,  
Monitor  
Petshop  
SignInOut,  
Monitor Home  
URL

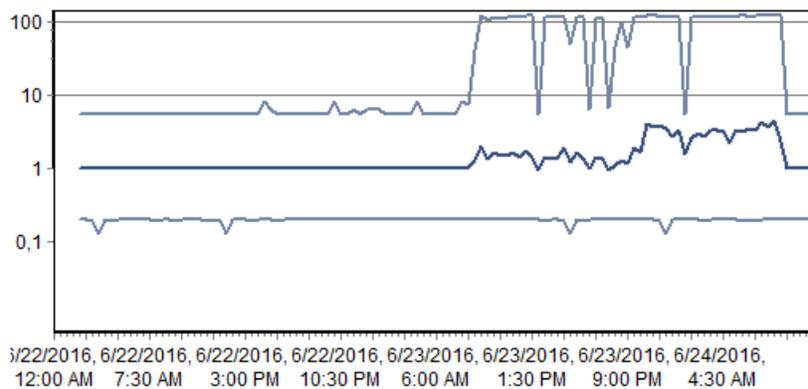
**Locations:**  
all

Average Page Time	Avg [s]	Min [s]	Max [s]	Count	Std.Dev.	Boundary 1	Boundary 2	Histogram
Page Time	1.44	0.12	130.00	15,669.00	6.76	85.80% (< 1.27)	87.68% (< 1.93)	
Document Download Time	0.73	0.12	130.00	31,338.00	6.60			
Server Busy Time	0.42	0.00	129.63	31,338.00	6.59			

**Page Time** (dark blue), **Document Download Time** (light blue), and **Server Busy Time** (medium blue) are color-coded and graphed separately. **Boundary 1** is color-coded yellow. **Boundary 2** is color-coded red.

To view the response time of a specific transaction, you must de-select all other transactions in the **Transactions** section of the project overview report. The example performance detail charts in this section are specific to average page times only. **Transaction Response Time**, **Custom Timer**, and **Custom Counter** charts show average, maximum and minimum values, rather than page time, document download time and server busy time.

The above example shows a linear chart. Linear charts present data in a natural, intuitive manner. Logarithmic data presentation is also available (shown below).



Maximum [s]  
Average [s]  
Minimum [s]  
No boundaries defined or merged boundaries differ.

- Download as CSV
- linear | logarithmic
- ▶ Show Page Details
- Update Report

**Transactions:**  
Monitor Home  
URL, Monitor  
Petshop Buy,  
Monitor  
Petshop  
Search,  
Monitor  
Petshop  
SignInOut

**Locations:**  
all

Average Page Time	Avg [s]	Min [s]	Max [s]	Count	Std.Dev.	Boundary 1	Boundary 2	Histogram
Page Time	1.44	0.12	130.00	15,669.00	6.76	85.80% (< 1.27)	87.68% (< 1.93)	
Document Download Time	0.73	0.12	130.00	31,338.00	6.60			
Server Busy Time	0.42	0.00	129.63	31,338.00	6.59			

With logarithmic charts you see much more detail in small values. Logarithmic data presentation is well suited to charts with measurements that do not vary much, but have abrupt spikes (large maximum measurements, small standard deviation). An example of ideal logarithmic chart use involves a value in a range that generally remains between "0.1" and "1," but has an abrupt spike to "100." On a linear scale, most of the values would be squashed in a small range along the bottom of the chart, one or two pixels in height. On a logarithmic chart, the values would receive 1/3 of the chart's space-"0.1 - 1" taking up the same space as "1 - 10" or "10 - 100"

To toggle between logarithmic and linear data presentation, click **Linear** and **Logarithmic**.

To view a tool tip that has details about a specific time interval, hold your cursor over a time interval of interest.

To progressively zoom into chart details (and thereby limit the amount of time covered by a chart), click a time interval of interest. To zoom out of a chart, use the **Less** ("-") magnifying glass link.

Data tables within chart details include:

- Average (Avg [s])
- Minimum (Min [s])
- Maximum (Max [s])
- Standard deviation (Std. Dev): An indication as to the stability of a measurement. If values are more or less always the same, the Std. Dev value will be relatively small. A large Std. Dev value indicates that a measure is quite variable.
- Boundary 1 (see [Boundaries](#))
- Boundary 2 (see [Boundaries](#))
- Histogram (see [Histograms](#))

The calendar and **Select range** features can be used to refine the time period covered by transaction **Details** pages.

Transaction performance data can also be downloaded in CSV (Comma Separated Values) format for use in spreadsheets and other charting engines.

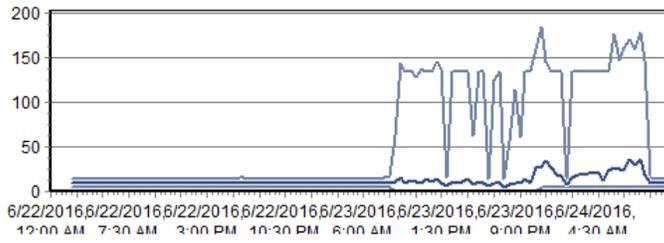
Click **Download as CSV** to download performance data to your hard drive in CSV format.

## Correlating Results

To compare different performance detail charts as well as numbers of accuracy and availability errors, select multiple measurements on the project overview report and click **Show comparison report** at the bottom of the page.

This generates a report with all performance details for the selected measurements on a single page and enables visual correlation of metrics. By default, data tables are hidden, so that more charts can be shown on each page. Click **Show data tables** to display data tables beneath the charts. See [Performance Detail Charts](#) for more information about what is available on this page.

Transaction Response Time (Client-side)



Maximum [s]  
 Average [s]  
 Minimum [s]  
 No boundaries defined or merged boundaries differ.

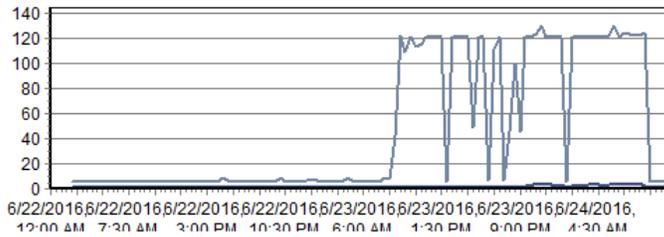
Download as CSV

[linear](#) | [logarithmic](#)

**Transactions:**  
 Monitor Petshop Buy, Monitor Petshop Search, Monitor Petshop SignInOut, Monitor Home URL

**Locations:**  
 all

Details for: Average Page Time (Client-side)



Maximum [s]  
 Average [s]  
 Minimum [s]  
 No boundaries defined or merged boundaries differ.

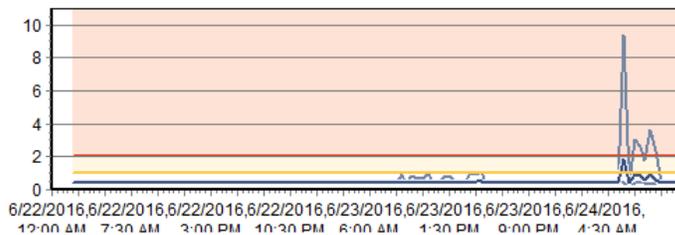
Download as CSV

[linear](#) | [logarithmic](#)

**Transactions:**  
 Monitor Petshop Buy, Monitor Petshop Search, Monitor Petshop SignInOut, Monitor Home URL

**Locations:**  
 all

Details for: TBuy-19-Order Information (Client-side)



Maximum [s]  
 Average [s]  
 Minimum [s]  
 Boundary 1  
 Boundary 2

Download as CSV

[linear](#) | [logarithmic](#)

**Transactions:**  
 Monitor Petshop Buy, Monitor Petshop Search, Monitor Petshop SignInOut, Monitor Home URL

**Locations:**  
 all

[Show Data Tables](#)

Upon returning to the **Client Health** report, you can select additional measures and click **Add to Comparison Report** to add the new measures to the comparison report.

**Automatic result correlation**

Because time is a critical factor when a production system is exposed to problems, Performance Manager offers options for automatically correlating results and streamlining root-cause analysis. This method involves correlating all results that experience a hit of the upper bound during a given time period (**Show Boundary #2 Violations**).

**Correlation based on boundary #2 violations**

The automatic correlation offered by Performance Manager is correlation based on boundary #2 violations. This feature facilitates root cause analysis by generating a comparison report that includes only those metrics that hit boundary 2 during the selected time frame. This is an easy means of gaining insight into the possible causes and related symptoms of performance issues.

Boundary #2 correlation reports can become unwieldy when too many metrics hit boundary 2. When no timers or counters have boundary 2 violations during the selected time frame, the report does not display; an informational dialog is displayed instead.

# Drilling Down Through Data

Performance Manager organizes health monitor results into a collapsible, tree-like structure that keeps detail at the top level to a minimum. Expanding and collapsing levels in the tree structure makes it easy for you to drill down through long lists of monitor results. Each custom/page timer in the report has a dedicated row. Current health ratings are shown for all results that have boundaries defined for them.

Isolating problematic components with the Performance Manager project overview report involves a drilling down process in which good performance data (time intervals, locations or browser types that do not report errors) is stripped away so that more detail regarding problematic elements can be revealed-and ultimately information that facilitates issue resolution is identified.

You can use the project overview report's **Select All**, **Deselect All**, and **Update Report** links to filter information and focus on areas of concern. Likewise the calendar and **Select Range** functions allow you to focus on periods of concern while disregarding periods of good performance.

To illustrate, consider that you have discovered a poor high-level health rating. On its own, this rating is not all that informative-it simply tells you that a problem exists somewhere in the system. By looking further down the report (i.e., by drilling down), you might discover that the cause of the poor **Health** rating is a poor **Availability** rating. Although that information offers some insight, it is still not all that helpful. You need to drill down further through the result data to get to the information that will help you resolve this issue.

Further down the report you might see that one of the transactions has been experiencing a problem while the other transactions have not been experiencing problems. Now you have some helpful information: one of the transactions is experiencing an availability problem. Still, the cause of the problem is not apparent. At this point you might deselect the properly functioning transactions to narrow the results down to the information that is related to the transaction that is experiencing an availability problem, and filter out all error-free performance data.

Finally, toward the bottom of the report, in the **Location** section, you might see that one of the locations-the location that is serving the transaction in question-has been having problems. Now you've got information you can act on; the cause of the overall poor health rating is not related to the application; it is related to a network problem at a specific monitoring location.

## Page Break-Down Analysis

While the discovery of correlations between results is the first step in root-cause analysis, detailed analysis of results is the second step. In addition to generating TrueLog files for front-end diagnostics, Performance Manager provides detailed page break-downs for page timers.

When you click a monitor you are taken to a detail page for that monitor. #1 boundaries (below which performance is good) are color-coded yellow. #2 boundaries (above which performance has failed) are color-coded red. Good, warning, and failure ranges are also color-coded in the charts.

Hold your cursor over a chart to display percentage break-downs of each measurement in a tool tip.

When you click the **Show Page Details** link you are presented with two more diagrams that offer a simple break-down of page timings, document downloads, etc. The third diagram offers more detailed analysis (connection handshake, SLL/TLS Overhead, HTML code, embedded objects, network details, etc).



**Note:** The **Show Page Details** link is enabled via the Silk Performer script. To enable the link, edit the active profile within Silk Performer and click the **Results** icon in the shortcut list on the left. Select the **Time Series** tab. Under **Detailed page timers**, click the **For the whole page** option button. For a URL checker monitor, use the **Detailed Page Timers** attribute.

The third (bottom) chart shows how time was spent during each time interval. It indicates what percentage of download time was spent sending requests, receiving information, etc. Hold your cursor over this chart to display a tooltip that tabulates the totals.

The results are displayed as a percentage, and not an actual value, because the overall value may be greater than the actual page download time. For example a browser will download a page from the server

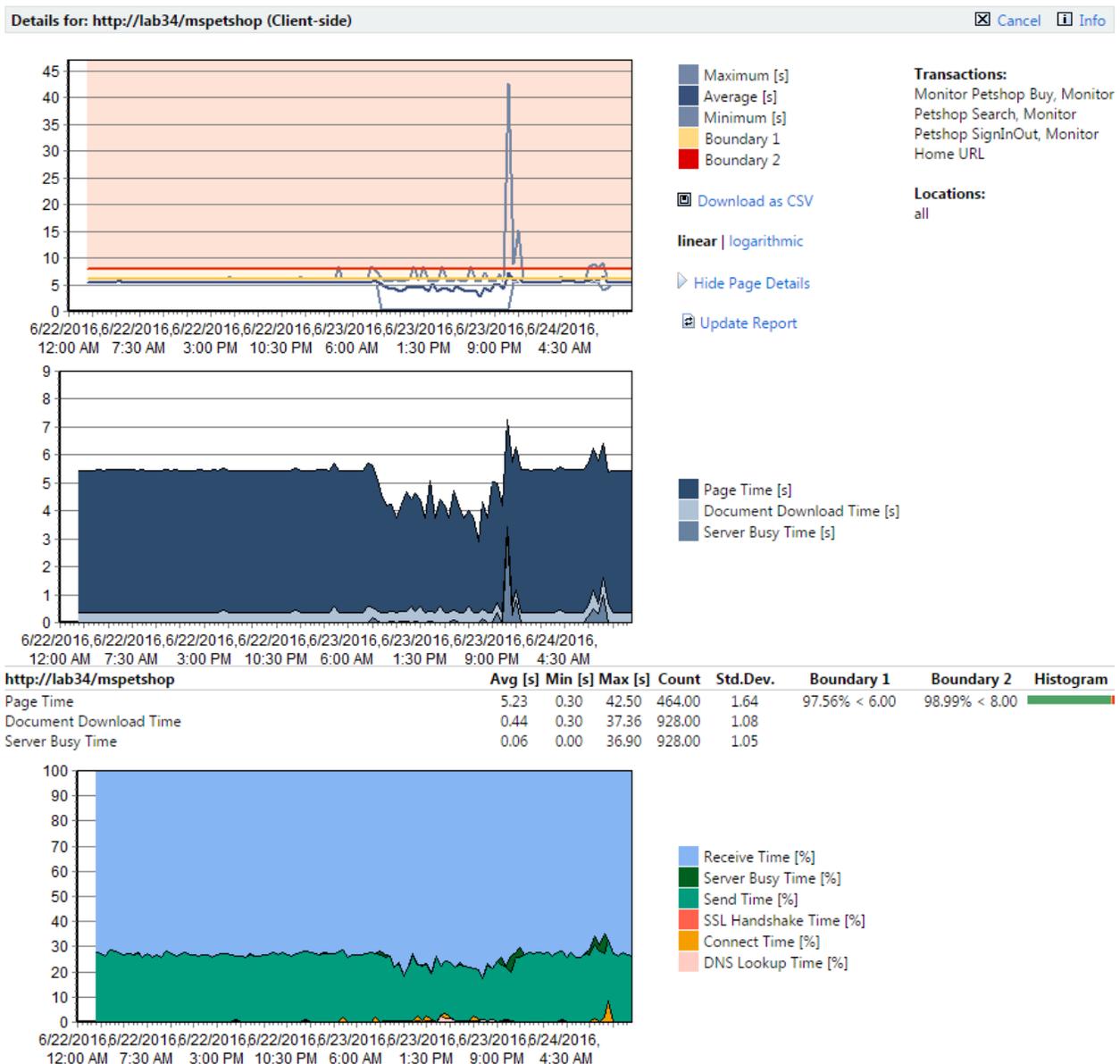
using several connections. If you added the sum for all of the measures displayed, the connect time, SSL handshake time, send time, receive time, server busy time, and DNS lookup time, the overall time can be larger than the actual download time (Page Time) by up to four times if four connections are used in parallel.

Dividing the total time by the number of connections will also not offer an accurate value as not all of the connections will be used throughout the entire download time and therefore the sum divided by the number of connections would be too small.

This means that it is only possible to display the times relative to the others, for example the receive time took 64.89% of the overall time when compared to the send time, which took 30.30% of the overall time.



**Note:** Boundaries are only included in charts when identical static boundaries are used, or when only one location is selected for analysis. This is because dynamic boundaries vary from location to location even when they measure the same transaction-averaging such boundaries would be confusing.



# Downloading TrueLog Files

Error listings offer the option of downloading `.wrt` files and `.xlg` TrueLog files. `.wrt` files contain information that is written with `write` and `writeln` Silk Performer script commands.

TrueLog information is tracked to support root-cause analysis of reported errors. TrueLog files can be downloaded as WinZip archives (extension `.zip`). "Zipped" TrueLog files are automatically extracted when opened with TrueLog Explorer and are recommended for use when limited bandwidth is an issue. You can click any monitor execution to display detailed diagrams that cover monitor-execution statistics such as page times, connection times, and handshake times. See the [Working with Silk Performer](#) section of the Introduction for details regarding TrueLog technology.

TrueLog files are only available when they are enabled in monitor settings. See [Adding Monitors](#) for details regarding enabling the writing of TrueLog files. Available TrueLog files are indicated with **Download Files** buttons.

To download a TrueLog file:

1. Click the corresponding **Download Files** button (grayed-out **Download Files** buttons indicate that no files are available) of the execution you want to analyze. The TrueLog File download page appears, listing the name of the file that is available for download (either `.xlg`, `.xlgz`, or `.wrt`).
2. Click the **Download** button of the file you want to download.

Alternatively, click the **Download as Zip** button of the file to download the file in WinZip archive format.



**Note:** To view TrueLog files you must have TrueLog Explorer installed on your system. For details regarding TrueLog Explorer, see the TrueLog Explorer Help.

# Incidents Log

The Incidents log lists all incidents that occurred during the selected time period, including both custom and system-detected incidents.

Incidents are comparable to error conditions. Incidents are raised when predefined conditions are met (for example, when performance drops below a certain minimum threshold for a certain length of time, and a service target violation is triggered).

Incidents are closed when the conditions that raise them are no longer active. Conditions that raise incidents are configured when monitors are configured. See [Transaction Conditions](#) for details.

See [Custom Incidents](#) for information regarding custom incidents.

See [Adding Rules](#) for information regarding defining service target incidents.

The **Incidents** tab (**Performance Manager > Monitoring > Incidents**) includes the following information for each incident:

- **From** - When the incident began
- **To** - When the incident ended
- **Duration** - How long the incident lasted
- **Name** - Name given to the incident (for system-detected incidents, this is simply the name of the transaction that led to the incident)
- **Severity** - The incident type (Error, Service Target Violation, Warning, or Informational)
- **Notification** - Whether a notification was executed in response to the incident. See [Configuring Rule Actions](#) for information regarding notifications.
- **Report** - Generates an incident report for the selected incident.

## Viewing all results

When the **Incidents** log contains more entries than can be included on a single page, entries are broken out over multiple pages. The **Result Page** number links (available at both the top and bottom of the page) allow you to jump to alternate result pages—simply click a page number. You can tell how many results out of the total you're currently viewing by the entries tag (for example, "entries 1-100 of 1920").

## Filtering incidents

The incidents log can become unwieldy when too many results are returned for a specified time period. You can filter down the incidents list to the incident names and types that are most relevant using the **Use filter "Any name"** and **"Any severity"** lists. Select an incident name and type from the lists and click **Update**.

## Sorting incidents

Incidents in the incident log can also be sorted by column (**From, To, Duration, Name, or Severity**). Click the column names or the **Ascending/Descending** arrow links to have results sorted by that column. To change sort-order-by-column from ascending to descending (or vice-versa) click the appropriate **Ascending/Descending** arrow link.

# Service Target Log

The **Service Target** log offers information regarding project-wide service targets, monitor service targets, and specific service target violations.

The calendar and **Select range** features can be used to refine the time period covered by the **Service Target** log.

## Project service target

The **Project Service Target** section of the **Service Target** page (**Performance Manager > Monitoring > Service Target**) includes heat fields that reflect overall service targets for the specified time period.

The **Project Service Target** section also includes information regarding:

- **Violations** - Number of service target violations
- **Average Duration** - Average duration of service violations
- **Uptime** - Percentage of time that the system was up and running
- **MTBF** - (Mean Time Between Failures) The average length of time from the end of one violation to the beginning of the next violation.

## Service target thresholds

The **Service Target Thresholds** section includes heat fields that reflect monitor-specific service targets for the specified time period.

This section contains single-line entries for each rule that you have defined using the **Service Target Violation** severity. Rules with this severity are the building blocks of your service-level management system.

The **Service Target Threshold** section includes the same information that is included in the **Project Service Target** section:

- **Violations** - Number of service target violations
- **Average Duration** - Average duration of service violations
- **Uptime** - Percentage of time that the system was up and running
- **MTBF** - (Mean Time Between Failures) The average length of time from the end of one violation to the beginning of the next violation.

## Service target threshold violations

The **Service Target Threshold Violations** section lists all service target violations that occurred during the specified time period.

Detailed information for each violation includes:

- **From** - When the violation began
- **To** - When the violation ended
- **Duration** - How long the violation lasted
- **Name** - The name of the violation (System detected incidents use the name of the service target violation rule. Custom incidents have user-defined names)
- **Severity** - The incident type (**Service Target Violation**, **Error**, **Warning**, or **Informational**)

## Viewing all results

When the **Service Target Violations** log contains more entries than can be included on a single page, entries are broken out over multiple pages. The **Result Page** number links (available at both the top and bottom of the page) allow you to jump to alternate result pages—simply click a page number. You can tell how many results out of the total you're currently viewing by the entries tag (for example, "entries 1-100 of 1920").

## Sorting service target violations

Violations can be sorted by column (**From**, **To**, **Duration**, **Name**, or **Severity**). Click a column name to have results sorted by that column. Clicking a column's white up-and-down arrows toggles results between ascending and descending order.

## Linking to the project overview report

Each service target violation has a **Report** link. Click the **Report** link to return to the project overview report with the selected violation centered in the selected time frame.

# Execution Log

The **Execution Log** lists all transaction executions that took place during the specified period of time.

The **Execution Log** includes the following information for each execution:

- **Timestamp** - When the execution took place
- **Transaction** - The transaction that was executed
- **Location** - The location from which the transaction was executed
- **Status** - The result of the execution (**Info**, **OK**, **Warning**, or **Error**)



**Note:** Here "Status" relates only to the success of the execution itself, not the status of the system under test. For that reason, executions that uncover errors may have a status of OK.

- **Message** - How long it took for the monitor to run and the interval at which monitors were executed
- **Results** - See *Execution Results* below for details.
- **Files** - See [Downloading TrueLog Files](#) for details.

## Viewing all results

When the **Execution Log** contains more execution entries than can be included on a single page, entries are broken out over multiple pages. The **Result Page** number links (available at both the top and bottom of the page) allow you to jump to alternate result pages; simply click a page number. You can tell how many results out of the total you're currently viewing by the entries tag (for example, "entries 1-100 of 1920").

### Filtering executions

The **Execution Log** can become unwieldy when too many results are returned for a specified time period. You can filter down the log to the transaction names, locations, and status that are most relevant to you using the **Use filter "Any transaction"**, **"Any location"**, and **"Any status"** lists. Select a transaction name, location, or status from the lists and click **Update** to filter results.

### Sorting executions

Executions in the **Execution Log** can be sorted by column (**Time Stamp**, **Transaction**, **Location**, **Status**, or **Message**). Click the column names or the **Ascending/Descending** arrow links to have results sorted by that column. To change sort-order-by-column from ascending to descending (or vice-versa) click the appropriate **Ascending/Descending** arrow link.

### Execution Results

To view detailed performance statistics for an execution's measures (**Avg**, **Min**, **Max**, **Count**, and **Std.Dev.**) click the corresponding **Results** link on the **Execution Log**.

To download historical execution results to your hard drive in CSV format, click **Download as CSV**.

# Creating Scripts for Silk Performance Manager

Silk Performer projects uploaded to Performance Manager may contain multiple scripts, user groups, and transactions. Transaction selection and the configuration of project attributes is done in Performance Manager. Therefore the need to download and modify scripts in Silk Performer is limited only to cases where scripts themselves need to be modified.

In this section, concepts are explained from both the Silk Performer and the Performance Manager perspective. Best practices related to the use of custom timers and counters in influencing performance-rate calculations are also detailed.

## Building Reusable Monitors

Several approaches are available for defining the aspects of Silk Performer projects that are utilized in Performance Manager monitors.

### Multiple Transactions

Multiple transactions can be defined in Silk Performer scripts. Transactions are associated with user groups. Silk Performer projects may contain multiple scripts.

In Performance Manager, when creating a new monitor, you select:

- A script (this step is omitted if the project contains only one script)
- A user group of the selected script (this step is omitted if the project contains only one user group)
- One or more transactions of the selected user group. The `begin` and `end` transactions cannot be selected.

The `begin` transaction is executed when the monitor is initialized.

On each scheduled execution, all selected transactions are executed once, independent of the transaction count defined in the script (i.e., Performance Manager uses its own workload model and ignores the model specified in the Silk Performer project).

The advantage here is that you can maintain several monitor scripts in a single Silk Performer project, using a different script or user group for each monitor.

The differences between using several transactions in a single monitor and creating a monitor for each transaction are as follows.

Several transactions in a single monitor	One monitor for each transaction
All transactions use the same schedule	Each transaction can have its own schedule
All transactions use the same process	A process is created for each transaction
The transactions are executed sequentially	Depending on the schedule, the transactions can run in parallel
All transactions share the same global variables	Each transaction has its own set of variables
All transactions use the same profile	A different profile can be assigned to each transaction

Several transactions in a single monitor	One monitor for each transaction
All transactions use the same project attributes	Different project attributes can be assigned to each transaction

## Profiles

Profiles can be used to control the behavior of monitors. After creation of a monitor, a profile may be selected from the list of defined profiles in Silk Performer. Of the settings defined in a profile, network bandwidth and browser type can be changed for client monitors while leaving the remaining settings of the profile unchanged. Also, the generation of output files (the writing of TrueLogs and the default output file) can be set to overrule the definition in the profile.

Note: Do not use a `WebSetBrowser` function call in the `begin` transaction, as this will override the setting defined in the Performance Manager GUI.

## Project Attributes

The best way to build reusable monitors is through the use of project attributes. Project attributes give you the option of parameterizing your script. Project attributes consist of a set of variables that can be defined from outside of Silk Performer projects, thereby defining an interface.

In Silk Performer, project attributes are defined in the **Project** menu. When creating or editing a monitor in Performance Manager, a list of configurable project attributes is displayed. The default values set in Silk Performer are shown by default in Performance Manager as well. The monitor then reads the attributes into variables during runtime.

Using this mechanism, variables such as user names, passwords, and account numbers, as well as host names, ports, and so on, can be defined during monitor creation. It is possible to create several monitors that share the same Silk Performer project, but use different project attributes.

## Timer Names

Timers (page timers, custom timers, etc.) are displayed in alphabetic order in Performance Manager reports. If you want to have timers sorted in a different order, for example, in order of sequence, you can rename the timers in the Silk Performer script so that an alphabetical sort order reflects the sort order that you want to see.

To sort page timers by sequence, rename your timers as follows:

```
"Start (http://...)" --> "01 - Start (http://...)"
"ExecuteLogin (form ...)" --> "02 - ExecuteLogin (form ...)"
"Page 1 (http://...)" --> "03 - Page 1 (http://...)"
```

# Configuring Infrastructure Monitors

Server context is required by infrastructure monitors to obtain relevant server parameters for monitoring. In Performance Manager, server context consists of IP address, host name, monitor settings, UNIX settings, and login credentials for SNMP, RPC, and Perfmon. Each of these parameters is required by the performance data collection engine. Server names don't need to be hard coded into scripts, they can be parameterized. All required server-context parameters are automatically imported into Performance Manager along with Silk Performer projects that are uploaded by Silk Performance Explorer. See [Requirements for Monitoring Windows Machines](#) and the Silk Performance Explorer Help for detailed information about setting up real-time monitoring.

## Requirements for Monitoring Windows Machines

Make sure that the following preconditions are met to be able to monitor the respective remote systems:

### Target machines with Windows Vista and newer client operating systems

The Remote Registry service must be running on the machine that you want to monitor. This service does not run by default on Windows Vista and Windows 7 machines.

### Target machines with Windows 2003 and newer server operating systems

- User is in Administrators group
- User is in the Network Configuration Operators and Performance Monitor Users groups
- User is in Users group, and the path `Software\Microsoft\Windows NT\CurrentVersion\Windows` is added to one of the following Network access settings:
  - Remotely accessible registry paths setting
  - Remotely accessible registry paths and sub paths under Local Policies/Security Options setting

This approach works well for tests with machines in different trusted domains and with users from those domains.

### Target machines with Windows XP operating systems

- **Simple File Sharing** is disabled
- READ access to the following files on the remote computer:
  - `%SystemRoot%\System32\Perfc009.dat`
  - `%SystemRoot%\System32\Perfh009.dat`
- READ access to the following registry keys on the remote computer:
  - `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg`
  - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib`

## Creating Silk Performance Manager Infrastructure Monitors

To access NT performance monitor data, the user who creates the infrastructure monitor with Silk Performance Explorer needs to be in the administrator group of the computer that you want to monitor. The execution server must also run under that user.

To set up an infrastructure monitor:

1. In Performance Manager, go to **Performance Manager > Infrastructure**.
2. Click **Add New Server** or select an existing server. The **Configure Server - Customize Server Attributes** page appears.
3. Edit the fields as required to access the server that you want to monitor, then click **Save**.
4. Start Silk Performer and create a **Silk Performance Manager - Infrastructure Monitor** project.
  - a) Click the **Outline Project** button on the Workflow bar. The **Workflow - Outline Project** dialog box opens.
  - b) Enter a name for the project in the **Name** text box and an optional description in the **Description** text box.
  - c) From the **Application Type** list, select **Monitoring > Silk Performance Manager - Infrastructure Monitoring**.
  - d) Click **Next**. Silk Performance Explorer opens and the **Data Source Wizard** dialog appears.
5. Click **Select from predefined Data Sources** and then click **Next**.
6. Expand the **Custom Data** folder, click **NT Performance Monitor Data**, and click **Next**.
7. In the **Hostname** field, specify the machine to be monitored.

8. *Optional:* In the **Alias** text box, specify the alias name.  
The alias must be a highly descriptive synonym for the monitored server. It is recommended that you group measures on a particular machine.  
For example, both WebLogic and IIS might be installed on the same computer. Both servers require monitoring, but the two performance measures must appear in separate menu trees.
9. In the **Username**, **Password** and **Domain** fields, specify a user who has administrative security rights. Click **Next**.
10. Examine your system for available performance counters and add them to the monitoring template. Click **Add** to add selected counters to the measure.  
To select multiple counters from the list, use **Ctrl+Click** or **Shift+Click**.  
The **Counter Usage** dialog appears.
11. Ensure that the **Is an average measure** check box is checked. Click **Next** and then click **Close**.
12. Check the check boxes for those measures that you want to include in the initial monitor view and then click **Finish**.
13. Back in Silk Performance Explorer, click the **Real-Time Monitoring** tab, and click **Export As Project** on the ribbon. The **Reuse Monitor Wizard** dialog appears.
14. If not already selected, select the **Hostname** for which the measures of the monitor chart are exported and click **Next**.
15. Verify the **Export parameters** and click **Finish** when you are done. The **Upload Project** dialog appears.
16. Make sure that the **Upload URL** points to the correct Performance Manager system (`http://<Performance Manager host>:19120/project_upload`) and click **OK**. If requested, enter valid login credentials to access Performance Manager.
17. In Performance Manager, go to **Performance Manager > Configuration > Monitors** and click **Add New Monitor**.
18. Under **Custom Monitors**, select the monitor that you just uploaded. The **Configure Monitor - Define Monitor Settings** page appears.
19. Edit the fields as required, but make sure that the selected **Server** corresponds to the server that you added at the beginning of this task. For detailed information, see *Adding Monitors*. Click **Finish** to save the monitor with the project-wide schedule.

As the processes in which transactions are executed are not stopped after scheduled executions, global variables within scripts remain valid. Therefore, it's possible to define variables in `begin` transactions and use them to remember values from previous executions. Using global variables for project attributes, variables can be initialized in `begin` transactions and used throughout scripts.

## Maintaining Monitors

When a monitor is edited, it is not possible to change the script or the user group. You can however change the subset of transactions, or modify the project attributes and profiles.

If another script or user group is needed, a new monitor must be created, as the new set of transactions will most likely be different.

For monitors using uploaded Silk Performer projects, projects can be downloaded and edited for maintenance purposes. When subsequently uploaded, it is possible to replace old Silk Performer projects with new projects using the `replace` package function. The new package must then be manually selected from the list of uploaded files. The new Silk Performer project must contain the same script and user group that was previously defined for the monitor.

A monitor refers to a particular script and the components of this script, for example the name, the profile, and others, and cannot execute another script. You can modify the code in the script and add new transactions to the script, but you cannot remove existing transactions from the project. Therefore, you can

only replace a project if the following components of the original project are also contained in the new project:

- All script names
- All names of transactions contained in one of the old scripts
- All user groups
- All profile names

To modify a monitor:

1. Download the Silk Performer project (**Administration > Files > File Pool**) to your local system for editing in Silk Performer. Click the **Download** button of the package you want to download.
2. Modify the Silk Performer project using Silk Performer.
3. Upload the edited project back to Performance Manager (**Administration > Files > File Pool**). Click the **Upload from Browser** button.
4. Edit the monitor in Performance Manager (see [Editing Monitors](#)) and click **Replace Package**.
5. Select the newly uploaded Silk Performer project.
6. Complete monitor configuration in Performance Manager.

## Influencing Performance Rate Calculations

All measured values are displayed in the report, even when they do not measure the performance of the monitored system. Therefore it is possible to use custom timers and counters that do not influence the performance rate. You can however view such values in reports and thereby gain additional data about the state of a system.

### Page Timers

Page timers influence the performance rate either by comparison to previously measured data of the same timer or by comparing the timer value to its boundary, as defined in the monitor settings. See [Calculating Health](#) for details.

In general, transaction response time is also used for performance rate calculation. To gain comparable values, think times are not counted toward transaction response time. Time spent in wait statements is also ignored.

### Transaction Response Time

If transaction response time is not to influence performance rate set in the client monitor settings to "display only" or "no results". In the first case, the response time will still appear in the report. This can be useful for transactions that query performance data rather than simulate business transactions. For example, the time it takes to read performance values from a statistics page, or the time that is required to query performance counters using the PDCE shouldn't be reflected in the performance health of a monitored system.

### Custom Timers

Custom timers can also be used for performance rate calculation. However by default they are shown in the report and do not influence the performance rate. If they are to influence performance, their name must begin with Sv\_. The name in the report shows the prefix to indicate which of the custom timers influence the performance rate and which do not.

Example:

```
HTIMER hMyTimer
...
hMyTimer = TimerCreate("Sv_LDAP Ping@"+sHost);
TimerStart(hMyTimer);
if WebLdapConnect(hLdap, sHost, nPort, nFlag) then
```

```

WebLdapBind(hLdap, NULL, NULL);
WebLdapDisconnect(hLdap);
end;
    TimerStop(hMyTimer)
    TimerDestroy(hMyTimer)

```

## Custom Counters

Custom counters can be used to indicate performance, accuracy, or availability of a system. By default, they are only displayed in reports and do not influence health rate. When a custom counter measures performance data, it must begin with the prefix `Pe_`. In such cases, boundaries should indicate whether large values are considered good or bad.

Example:

```

MeasureSetBound("Pe_CPU", MEASURE_COUNTER_CUSTOMCOUNTER, 1,
float(nBoundCpu1));
MeasureSetBound("Pe_CPU", MEASURE_COUNTER_CUSTOMCOUNTER, 2,
float(nBoundCpu2));
MeasureIncFloat("Pe_CPU", fValue, "%");

```

For custom counters to influence accuracy, the prefix `Ac_` is used. The prefix `Av_` creates a custom counter that influences the availability rate. When such a counter is set, the monitor reports a problem in the accuracy or availability of the monitored site.

Example:

```

MeasureInc("Av_Host not reachable");

```

## Writing Result Files

While TrueLog files, default output files, and other default Silk Performer result files are stored automatically, other files may also be written.

For a file to be sent from the execution server to the application server, it must be written to the default result folder on the execution server.

Example:

```

GetDirectory(DIRECTORY_RESULT, sDir, sizeof(sDir));
FOpen(hFile, sFileName, OPT_FILE_ACCESS_READWRITE, OPT_FILE_CREATE);
FWrite(hFile, sOutput, strlen(sOutput));
FClose(hFile);

```

## Writing Action Essentials

An action Essential is a special type of Essential that is executed by Performance Manager when incidents begin and end.

To set up an execution server for action Essentials, refer to the Performance Manager Administration Help.

An action Essential is a Silk Performer project that is used to enhance the list of available actions that can be linked to a rule. The handling of scripts, user groups, and transactions is the same as with monitoring Essentials, except that only a single transaction may be selected.

Project attributes can be used; the following attributes must be present:

Type	Indicates whether the incident started (1), ended (0), or the test action link was used to start an action (3).	Number
------	---	--------

Project	The project to which the action belongs.	String
Severity	Severity of the incident.	String
Rulename	Name of the rule that triggered the action.	String
Expression	Expression of the rule that triggered the action.	String
Rise_time	Start time and date of the incident.	String
Fall_time	End time and date of the incident.	String
Duration	Duration of the incident.	String
ReportLink	Link to the report showing the incident.	String

Performance Manager automatically sets all values for these attributes when rules are evaluated, so there's no need to define them when configuring an action. Additionally, other project attributes may be used. These must be set when the action is created.

To write a custom action Essential, it is recommended that you use the Essential action template that is pre-installed with Performance Manager: Go to the Essential administration page and download the action template. Open the Essential package with Silk Performer and modify it. The sample shows you how to use an action Essential to send email; a custom project attribute is thereby added. Modify this according to your needs and upload the action template to Performance Manager. Create a new Essential from the uploaded template and define it as an alerting Essential.



**Note:** This task needs to be done by a user with the **Administrator** role.

# Calculating Health

This section explores how results measured by Performance Manager agents are used to calculate health rates. The processes for calculating and interpreting health rates are also described in detail.

## Health Rates

In contrast to monitoring tools that simply present measurement data, Performance Manager contextualizes monitor results with health rates. Health rates enable analysts who have little familiarity with monitored applications to readily evaluate monitor results. They also assist experienced analysts in pinpointing relevant measurements.

The primary challenge associated with interpreting monitor reports is that when you're not familiar with their format, you need to spend time analyzing their meaning. Even when you are able to efficiently interpret a report format associated with a certain project, when you view a different report format designed for a different project you must once again evaluate what is relevant and what can be ignored. In response to this challenge, Performance Manager converts all of its measurement data into percentage rates. All relevant data is thereby converted into abstract percentage values that are aggregated into single high-level values, or health rates, that reflect overall project readiness.

The benefit of health rates is that they offer analysts a short cut for evaluating project health and directing development efforts. If the overall health rate of a project is solid, there is no need for further analysis. Health rates have values between "0" and "100" (0 being the worst, 100 being the best) and are independent of projects, the amount of data analyzed, and the frequency of individual measurements.

Because high-level health rates are the aggregate of low-level health rates, analysts have the option of reversing rate calculations to determine how specific low-level rates influence overall rates. Such causal analysis can be used to "drilldown" to specific low-level data that negatively affects overall rates, thereby pinpointing the system components that have a negative impact on system health. All the while, measurements that fall within acceptable ranges are ignored.

Because low-level health rates reveal the fitness of actual measurement values, analysts don't need to understand the significance of measurement values themselves. For example, without familiarity with a certain monitored application, it would not be readily apparent whether a business transaction that takes 15 seconds is fast or slow. A health rate of "95%" however is readily understood to be a healthy rate.

There are several aspects of monitored results. The first aspect is result type (response time, error message, counter, etc.). Each monitored result is also associated with a certain transaction, which runs on a specific location. Once rates are calculated they are aggregated into result type, and ultimately used to generate the three health-dimension rates: Availability, Accuracy, and Performance.

The first step in evaluating poor health rates is to examine the associated health dimensions. Say for example that a certain monitor or agent encounters problems with a server under test. In such an instance, only the relevant data related to the server experiencing problems would be presented. In this way, Performance Manager simplifies the process of identifying the causes of detected problems.

To calculate how expected measurement values correlate with the health rate scale of 0-100 (without the input of an expert who is familiar with the capabilities of the system under test), one has to rely on comparisons of past measurement values. In the same way that an athlete estimates his fitness by comparing his recent achievements to past performance, Performance Manager compares measured values to measurement values recorded in the past. This method evaluates what is possible, but it does not offer an indication of maximum performance potential. Load testing and benchmarking are better suited for such assessments. Note that with this calculation approach, it must be assumed that production systems are performing well and that variations from benchmark rates warrant analyst scrutiny.

# Health Dimensions

Overall health values are influenced by the three health dimensions availability, accuracy and performance.

Each of these health dimensions and the overall health rate itself are represented by values in the range of 0-100 (the higher the value, the better the health of the system).

Health values aren't always expressed as percentages (10%, 20%, 30%, etc); only availability and accuracy values are expressed as percentages. Performance and overall health values are expressed as absolute rates (10, 20, 30, etc.).

Health dimension values are calculated after each monitoring transaction. For individual transaction runs, both availability and accuracy are calculated to be either 0% or 100% (intermediate values aren't applicable for these health dimensions).

Note that accuracy and performance values aren't always calculated:

- Accuracy is calculated only when a monitored system is available (Availability = 100%).
- Performance is calculated only when a monitored system is available and accurate (Availability = 100%, Accuracy = 100%).

If these standards aren't met and performance values are based on a faulty system, when other values are compared to the optimum performance of the system, misleading results will be returned.

When you review the health dimension values offered by Performance Manager, you normally review values that have been calculated based on multiple monitoring transactions. So, when evaluating health dimension values, keep in mind that the health dimension value for a set of monitoring transactions is equal to the average value of all of the corresponding and existing health dimension values of all individual monitoring transactions.

## Availability

Availability is the most basic health dimension. It measures the percentage of time during which a monitored system is available to a subset of selected data. The availability rate provides information regarding whether a monitored system is running and whether it provides basic responsiveness to client requests.

A system is judged available when a monitoring transaction testing a system completes without detecting an error. Most errors indicate that a system is not available. Exceptions include those errors that indicate that a system is available, but not working correctly.

If several monitors are supervising a system and some of those monitors detect that the system is not available while other monitors detect that the system is available, the availability of the system will be rated near the middle of the 0% - 100% scale.

The following types of transaction results will cause Performance Manager to consider a monitored system to be unavailable:

- BDL transaction messages that use `SEVERITY_ERROR`. Though not all transaction error messages cause Performance Manager to set Availability to 0. There are some types of messages that influence Accuracy.
- Custom counters created with BDL functions `MeasureInc()` and `MeasureIncFloat()`. When a transaction creates a custom counter with a name that begins with the prefix `AV_`, Performance Manager sets Availability to 0. A custom counter with the name `AV_Server application not running`, for example, would result in an Availability rating of 0. The `Server application not running` Availability message would appear in the Performance Manager GUI.

## Accuracy

The accuracy rating for monitored systems is calculated only after systems are determined to be available.

Accuracy rates reflect whether monitored systems work as designed and if information transmitted to clients is correct. Useful functions that can be evaluated to determine accuracy include link checking, content validation, title validation, and response data verification. If a monitoring script contains customized functions that are used to ascertain system accuracy, these functions will also be factored into the accuracy rate.

Accuracy ratings go far beyond simply checking for availability. A server may be available even when the application it hosts isn't responding. Likewise, dynamic pages may be corrupt, database queries may produce empty result sets, and warehouses may run short of stocked merchandise. The simple checking of availability will not alert one to such failures. Only complex transactions that compare results to benchmarks detect such problems.

The following transaction results can cause Performance Manager to consider a monitored system to be inaccurate:

- The following BDL transaction error messages are associated with Accuracy:
  - HTTP errors that are not of the 5xx kind (for example, HTTP 502 - error response received from gateway is associated with Availability)
  - Errors resulting from verification functions
  - Custom errors created using the `RepMessage()` BDL function
- Custom counters created using the `MeasureInc()` and `MeasureIncFloat()` BDL functions. When a transaction creates a custom counter with a name that begins with the prefix `AC_`, Performance Manager sets Accuracy to 0. For example, a custom counter called `AC_Incorrect sort order` would result in an Accuracy rating of 0. An `Incorrect sort order Accuracy` message would then appear in the Performance Manager GUI.

## Performance

Once a monitored system is judged to be accurate, the performance health dimension of the system is calculated. Performance rates are based on sets of timers and counters that are created via BDL transaction system-monitoring.

First, each measurement that influences performance (for example, page-load time for a "Welcome to the shop" home page) is transposed into a performance value in a process called normalization. Secondly, the average of the performance values is used to define the performance rate for the corresponding transaction run.

The following transaction measures are considered in calculating Performance:

- Transaction Response Time
- Average Page Time of Web pages - When a transaction creates timers for individual pages, the Average Page Time is ignored when calculating Performance.
- Page load times for individual Web pages
- Custom timers whose names begin with the prefix `SV_`
- Custom counters whose names begin with the prefix `PE_`

These measures can also be configured in the GUI and in the Boundary Editor. See [Configuring Silk Performance Manager](#) for details.

### Example

Assume, we have a transaction called `CheckMyServer`. In the GUI we define the performance setting as follows:

- Transaction Response Time: display only
- Page Timers: performance rating
- Custom Measurements: performance rating

The transaction `CheckMyServer` calculates the following measures:

- Transaction Response Time: 8.2 secs.
- Average Page Time: 16.5 secs.
- Page Time for the Login Web page: 3.8 secs.
- Custom counter `PE_Handles`: 296
- Custom timer `InitTransaction`: 0.82 secs.

First Performance Manager calculates Performance values for some of these measures, as described below (the values specified here are for demonstration purposes only):

- Transaction Response Time is set to "display only" in this example and is therefore ignored.
- Average Page Time is also ignored because there is a page time available for the "Login" page.
- Login Web page: Performance 92
- Counter `PE_Handles`: Performance 32
- Timer `InitTransaction` is ignored because it doesn't begin with the prefix `SV_`

Secondly, the average of the calculated Performance values is used to define the Performance rate for this run of the transaction. This results in a Performance rate of 62 for the run.

### Normalization

There are two options for converting individual measurements into rates ranging from 0 to 100:

- With the assistance of an expert who is familiar with the capabilities of a system, good and bad measurement values can be identified.
- If no such expertise is available, values can be compared to previous measurements to determine if they are relatively good or relatively bad compared to past performance.

There is no difference between calculating performance rates for timers and calculating performance rates for counters. Both are treated the same, thereby enabling health rate comparisons.

### Normalization using boundaries

In the first scenario (having an expert specify good and bad measurement values) static boundaries, defined using the Boundary Editor, are used to define comparison values. Therefore, there are two values,  $b_1$  and  $b_2$ .

Boundary  $b_1$  may be considered the better value (i.e., the "warning level").  $b_2$  represents a more critical value (i.e., the "error level"). So if lower values are better than higher values, one must define  $b_1$  as being lower than  $b_2$ . If higher values are better, one must define  $b_2$  as being lower than  $b_1$ .

If the two boundaries are equal (or if they aren't set), there is no way to tell if higher or lower values are better. Performance Manager can't normalize such user-bound measurements and therefore such values do not influence performance rate.

When an expert provides two values for comparison, the performance rates that correspond to the values must be specified. This is defined using the Performance Manager `SVAppServerHomeConf.xml` configuration file and is the same for all measures in all projects. Default values specify that a measured value equal to  $b_1$  results in a performance rate of 80, while  $b_2$  corresponds to a performance rate of 50.

Other values are evaluated as follows: if a value is beyond  $b_2$ , the performance rate is determined to be 0. The reason for this is that Performance Manager considers  $b_2$  to be the threshold of unacceptable values. All values beyond this threshold are treated as performance failures of the monitored application.

Between  $b_1$  and  $b_2$  values are interpolated exponentially. This also applies to values beyond  $b_1$ , unless the interpolated value is higher than 100. In such cases the performance rate is determined to be 100.

The following chart illustrates user-bound normalization.



The formula that is used to interpolate the rating curve is an exponential curve through the two points  $(b_1, R_1)$  and  $(b_2, R_2)$ :

$$R_1 \cdot \left( \frac{R_1}{R_2} \right)^{\frac{x-b_1}{b_1-b_2}}$$

The entire formula is:

$$R = \begin{cases} \min \left( R_1 \cdot \left( \frac{R_1}{R_2} \right)^{\frac{x-b_1}{b_1-b_2}}, 100 \right) & x \leq b_2 \\ 0 & x > b_2 \end{cases}$$

Where  $b_1$  and  $b_2$  are the boundaries,  $R_1$  and  $R_2$  are the corresponding performance rates, and  $x$  is the measured value.

### Normalization based on past measurements (automatic bounds)

If the boundaries in a script are not defined, or if values are not useful for monitoring, Performance Manager can calculate performance rates by comparing measured values with previously recorded measurements. This approach offers relative judgments on the state of a system, rather than absolute judgments. When systems perform within their normal range, there is no need to raise warnings. If normal performance is bad, monitoring won't help much and systems must be redesigned or re-scaled. Therefore Performance Manager looks for exceptional values-exceptionally high values in most cases. If a custom timer or a custom counter is used, the boundaries defined in the script are used to define if a high value is good or bad. Apart from that however, values are ignored. When the boundaries of a custom timer or counter happen to be equal (for example, both are set to 0), the lower value is considered to be better.

In order to judge if measured data can be considered normal or out of the ordinary, Performance Manager uses the average and the standard deviation of historical value. A value has to lie far away from the average values, where far away means much more than the standard deviation. The idea therefore is, to use a band around the average, where the width of the band is derived from the standard deviation. That band defined which values are considered to be normal. For example for a rather stable system the measured values will lie closely together. The standard deviation will be small and the band within the future values are expected to lie will be narrow. A small deviation from the average will already suffice to raise suspicion as this is not common in this scenario. Another example could be a Web site where the page time vary depending on other network traffic and other influences. Here the standard deviation would be rather high and only exceptionally high response times, like time outs, would lie outside of this band. Thus, false alarms caused by normal variation of the throughput would be eliminated.

So in this approach, the boundaries  $b_1$  and  $b_2$  are calculated using the average ( $\mu$ ) and standard deviation ( $s$ ) of previously measured data. In general, the average and standard deviation is calculated using all historical data beginning with the creation of the monitor. This results in rather stable boundaries after an initial calibration period. The length of this period depends of the schedule interval and the distribution of the measured data. Generally, after about 100 measure points, that is after a day using a 15 minutes schedule, the boundaries should be calibrated.

By default, all data since the monitor was created is used for calculating the dynamic boundaries. Sometimes, the normal behavior may change, because of hardware changes like new servers or a better network connection, or because of software updates with a better overall performance. In these cases it would take a rather long time for the boundaries to readjust. Therefore it is possible to reset the dynamic boundaries and restart the calibration process beginning with the current date. See [Configuring Static Boundaries for Performance Values](#) for a description of this feature in the Web GUI.

Historical data defines the dynamic boundaries  $b_1$  and  $b_2$  with this approach, however the corresponding performance rates of  $R_1$  and  $R_2$  are still defined in the configuration file.

To calculate dynamic boundaries, the average ( $\mu$ ) and standard deviation ( $s$ ) of previously measured data are used, along with two factors that are defined in configuration files along with performance rates,  $R_1$  and  $R_2$ . By default, these two factors are  $f_1 = 1$  and  $f_2 = 6$ .

Therefore,  $b_1 = \mu + f_1 \cdot s$  and  $b_2 = \mu + f_2 \cdot s$ .

In the case of custom timers and counters, if the boundaries defined in a script indicate that higher values are better than lower values, then  $b_1 = \mu - f_1 \cdot s$  and  $b_2 = \mu - f_2 \cdot s$ .

This formula has a slight disadvantage if there are only a few values available. If the first data points lie closely together the standard deviation is small and the average may still vary. In order to compensate this, a factor is introduced to widen the band of allowed values by pushing  $b_1$  and  $b_2$  further away from the average.

The final formula is  $b_i = \mu \pm f_i \cdot (s + 2|\mu|/(N+1))$ , where  $N$  is the number of data points used to calculate the average and the standard deviation. The term  $2|\mu|/(N+1)$  gets smaller with every new value until it does not influence the boundaries any longer.

## Overall Health

Overall Health is the highest-level health rating and is an aggregate of Availability, Accuracy, and Performance ratings.

Once a transaction judges a system to be available and accurate, the Overall Health rating of that transaction is calculated based on the Performance rating.

If a system is not working accurately or is unavailable (either Accuracy or Availability equal 0), the Overall Health rating will be 0.

As Overall Health is based on the Performance health dimension value, it is an absolute rate not expressed in percentages (for example, 50, 80, etc.).

*Overall Health = Performance*

## Conclusion

Health rate calculation offers a single measurement that reflects overall system health, enabling analysts to detect high-level problems at a glance.

By reversing health rate calculations, analysts can readily determine if errors are availability problems, accuracy problems, or performance problems. Ultimately, analysts can drill down to the relevant measurement data that is responsible for specific errors.

Health rates do not replace data measurements; they simply contextualize measurements and, as they are expressed as percentages (Availability, Accuracy) and absolute rates (Overall Health, Performance), they can be aggregated to provide concise status overviews of systems under test. Specific measurement values remain important for in-depth analysis, experienced analysts, service target agreements, and notifications.

# Reports

This section explains how to generate reports with Performance Manager, download report templates, edit report parameters, and create new reports based on pre-installed templates. It also includes descriptions of all default report types that come pre-installed with Performance Manager.

For information about editing report templates and creating custom report templates via BIRT RCP Designer and MS Excel, see the *Silk Performance Manager Administration Help*.

## Creating Reports

Performance Manager offers reports that quickly and easily transform data into intuitive charts and graphs.

Reports are created using either BIRT RCP Designer, an open-source, Eclipse-based report tool, or Microsoft Excel report templates. Performance Manager is tightly integrated with BIRT RCP Designer to make it easy for you to generate reports on performance monitoring data. The reporting functionality in Performance Manager is highly customizable. Numerous pre-installed report templates provide out-of-the-box options for a wide range of reporting needs. Simple GUI-based tools allow you to edit the pre-installed reports and create reports of your own. For users with SQL knowledge, there is virtually no limit to how data can be queried and presented in custom reports.



**Tip:** If a blank report is generated, the cause may be that there are not any data in the project you selected, or you may not be connected to the appropriate Performance Manager database. Reports are not available offline unless your Performance Manager database is accessible locally.

## Creating New Reports

To create a new report:

1. In the menu, click **Performance Manager > Reports**.
2. In the **Reports** tree, select the folder in which you want the new report to display.  
This determines where the report is stored in the directory tree.
3. Click  on the toolbar. The **Create New Report** dialog box opens.
4. Type the name of the new report.  
This is the name that is displayed in the **Reports** tree.
5. Check the **Share this report with other users** check box if you want to make this report available to other users.
6. Type a description for the report in the **Description** field.
7. In the **Timeout [s]** field, type the maximum time period in seconds that Performance Manager should wait for SQL queries to complete.
8. From the **Default tab** list, select the tab that you want to be directed to when you select this report from one of the context-sensitive report lists.
9. Select the corresponding result type from the **Result category** list.  
This setting specifies the database table and view that is to be filtered for the report. Each result type offers a set of selection criteria. Based on the result type you have selected, specify an appropriate **Selection criteria** for your report. These criteria typically group properties based on a view or some other intuitive grouping, for example result properties.
10. From the **Property** list, select the property that is to be filtered on.  
For some selection criteria, properties are dynamic.

11. Select an **Operator** for the query.

The available operators depend on the property. Example operators are =, not, like, and not like. Strings are always compared lowercase. Allowed wildcards for strings are "\*" and "?", where \* matches any characters and ? matches exactly one character.

12. Select or specify the **Value** that the query is to be filtered on.

For date-based properties, the **Value** field is replaced with a calendar tool that you can use to select a specific date.

13. *Optional:* To add an additional query string to this report, click **More**. An existing query string can be deleted by clicking the string's **Delete** button. When multiple query strings are defined, **AND** and **OR** option buttons are displayed next to **More**. Use these option buttons to define if the queries should be considered cumulatively, or if only one query string's criteria needs to be met.

14. Click **Next** to configure report columns on the **New Report** dialog box.

15. Click **Add Columns**. The **Add Columns** dialog box lists all available report columns.

16. Select the columns that you want to have included in the report and click **OK**.

You can select multiple columns with **Ctrl+Click**.

The selected columns display in tabular format on the **New Report** dialog box.

17. *Optional:* Configure how each report column is to be displayed. For each column, specify a sort direction, ascending, descending, or unsorted, using the up/down arrows in the **Sorting** column.

18. When multiple columns are selected for sorting, a list box is displayed in the **Sort Order** column that allows you to more easily edit the column-sort order. Set these numbers as required.

19. Give each column an **Alias**.

This is the name by which each column will be labeled in the generated report.

20. With grouping, you can take advantage of SQL aggregation features, for example when selecting a number of elements or querying a total sum of values. Check the **Group by** check box to specify that SQL group by functions are to be applied.

21. Columns that are not selected for SQL group by functions are set to aggregation by default, which means a single aggregate value is calculated. From the **Aggregation** list, select the appropriate aggregation type.

The following types are available:

- Count
- Sum
- Average
- Minimum
- Maximum

22. The **Actions** column enables you to move column listings up and down in the view, or to delete a column.

23. Click **Finish** to complete your new report.

## SQL Functions for Custom Reports

To assist in writing advanced queries, placeholders are available for each function. Function placeholders are replaced with SQL code upon execution. Functions are used like parameters, but their names have a \$ (dollar symbol) as a prefix. Unlike parameters, placeholders are defined report elements that cannot be customized per execution.

The following table lists all available function placeholders:

Function	What it does	Example
\$TODAY	Returns the current systemdate on the database server. You can also	CreatedAt > \${\$TODAY}

Function	What it does	Example
	write <code>\$TODAY-1</code> for yesterday or <code>\$TODAY-7</code> for a week ago.	
<code>\$DATE(column)</code>	Returns the date but not the time.	
<code>\$DATE('string')</code>	Converts the given string to a database date.	<code>CreatedAt &gt; \$ { \$DATE( '01/10/2005' ) }</code>
<code>\$DAYS[p1;p2]</code>	Calculates the difference in days between the two given parameters. The two parameters can be a column within the table/view or <code>\$TODAY</code> .	The following example returns the rows created within the last week: <code>{ \$DAYS[CreatedAt; \$TODAY] } &gt; 7</code>
<code>\$WEEK(param)</code>	Returns the week-number of the given parameter, which can be <code>\$TODAY</code> or a column.	
<code>\$MONTH(param)</code>	Returns the month of the year as a number of the given parameter, which can be <code>\$TODAY</code> or a column.	
<code>\$YEAR(param)</code>	Returns the year as a number of the given parameter, which can be <code>\$TODAY</code> or a column.	
<code>\$USERID</code>	The ID of the currently logged in user.	
<code>\$USERNAME</code>	The name of the currently logged in user.	
<code>\$PROJECTID</code>	The ID of the currently selected project.	
<code>\$PROJECTNAME</code>	The name of the currently selected project.	
<code>\$REPORTNAME</code>	The name of the currently selected report.	
<code>\$REPORTID</code>	The ID of the currently selected report.	

## Writing Advanced Queries with SQL

Advanced reports can be created through manual SQL coding. Virtually any reporting option is available if you know the database schema. Clicking **Advanced Query** hides the query string list boxes and opens the **Report data query** field in which you can insert existing code or write new SQL code.

 **Restriction:** The SQL statement `select top` is not supported.

One approach is to begin query-string construction using the list boxes as outlined in [Creating New Reports](#). If the report criteria are valid, the equivalent SQL statement will be generated and displayed, and then move to advanced mode for further modifications.

 **Note:** If you switch from advanced mode back to simple mode the changes you made within the code will be lost.

To write an advanced query directly in SQL:

1. In the menu, click **Performance Manager > Reports**.
2. In the **Reports** tree, select the folder in which you want the new report to display. This determines where the report is stored in the directory tree.

3. Click  on the toolbar. The **Create New Report** dialog box opens.
  4. Type the name of the new report.  
This is the name that is displayed in the **Reports** tree.
  5. Check the **Share this report with other users** check box if you want to make this report available to other users.
  6. Type a description for the report in the **Description** field.
  7. Click **Advanced Query** to open the **Report data query** field. Insert previously written code or write new code directly in the field.  
The **Insert placeholder** list assists you in editing the SQL queries with pre-defined function placeholders. For details, see [SQL Functions for Custom Reports](#).
-  **Note:** If you manually edit SQL code for the query, click **Check SQL** to confirm your work.
8. Click **Finish** to save your settings.

## Sample Report

Below is the code of a pre-installed report called **Project Status Report**. By default, this report displays overall health values for all projects during a specified time span, from a beginning time (Begin|'01/01/2010 00:00'}) to an end time (End|'01/01/2030 23:59'}).

Detailed values for availability, accuracy, and performance are included:

```
(tsd.MeasureName = 'Overall Health'
OR tsd.MeasureName = 'Availability'
OR tsd.MeasureName = 'Accuracy'
OR tsd.MeasureName = 'Performance')
```

An aggregation level is set:

```
WHERE tsd.AggregationDescription = ${Aggregation|'Week'}
```

Here is the complete SQL code for the report:

```
SELECT projects.ProjectID_pk ProjectID, projects.ProjectName, tsd.MeasureName
MName,
SUM(tsd.ValCount) CountSeriesTime, SUM(tsd.ValSum)/SUM(tsd.ValCount) AvgValue

FROM SCC_Projects projects
INNER JOIN (SELECT DISTINCT pg.ProjectID_pk_fk
FROM SCC_Projects_Groups pg
INNER JOIN SCC_UserGroupRoles ugr
ON pg.GroupID_pk_fk = ugr.GroupID_pk_fk
WHERE ugr.UserID_pk_fk = ${USERID}) p2
ON (projects.ProjectID_pk = p2.ProjectID_pk_fk)
INNER JOIN SV_V_Monitors_TimeSeriesData tsd
ON projects.ProjectID_pk = tsd.ProjectID

WHERE tsd.AggregationDescription = ${Aggregation|'Week'}
AND (tsd.MeasureName = 'Overall Health'
OR tsd.MeasureName = 'Availability'
OR tsd.MeasureName = 'Accuracy'
OR tsd.MeasureName = 'Performance')
AND tsd.SeriesTime >= ${DATETIME(${pmResults_Begin|'01/01/2000 00:00'})}
AND tsd.SeriesTime <= ${DATETIME(${pmResults_End|'01/01/2020 23:59'})}

GROUP BY projects.ProjectID_pk, projects.ProjectName, tsd.MeasureName

ORDER BY projects.ProjectID_pk ASC
```

# Working with Sub-Reports

## Adding Sub-Reports

To aggregate the results from multiple reports into the currently selected report, you can add sub-reports. When adding a report as a sub-report, the result columns and rows of the sub-report are concatenated to the results of the selected report.

To add a report as a sub-report:

1. In the menu, click **Performance Manager > Reports**.
2. Select a report in the **Reports** tree.
3. On the **Properties** tab, click **Add Subreport**.
4. On the **Add Subreport** dialog, select the sub-report you want to have appended to the current report by selecting it from the **Reports** tree.
5. Click **OK** to complete the addition of the sub-report. Sub-reports appear on the associated report's **Properties** tab in a section called **Subreports**.

## Deleting Sub-Reports

To delete a sub-report:

1. In the menu, click **Performance Manager > Reports**.
2. Select the report in the **Reports** tree that has the associated sub-report that you want to delete.
3. Click the **Properties** tab.
4. Click **X** in the **Action** column of the sub-report that you want to delete.
5. Click **Yes** on the confirmation dialog box to confirm the deletion.

# Report Templates

Performance Manager report templates render report data into formats that meet your specific needs.

Templates can take the form of Excel spreadsheets, BIRT RCP Designer templates, XML, or CSV files.

## Uploading Report Templates

To upload a template from your local system:

1. In the menu, click **Performance Manager > Reports**.
2. Select the report to which you want to associate the template.
3. Click the **Report** tab.
4. Click the **Click here to upload a new report template** link to open the **Upload Report Template** dialog box.
5. Give the template a meaningful **Name** and **Description**.
6. In the **Projects** list box, select the project to which you would like to make the template available or select **All Projects** to have the template associated with all projects.
7. Browse to and select the template on your local system.
8. Click **OK** to upload the template.

# Downloading Report Templates

Downloading Performance Manager report templates to your local system enables you to edit them through BIRT Report Designer or Microsoft Excel. After you download and edit a report, you can upload it to make it available to other users. For details see the related [Uploading Report Templates](#) procedure.

1. In the menu, click **Performance Manager > Reports**.
2. Select a report that utilizes the template you want to modify from the **Reports** tree.
3. Click the **Properties** tab.
4. Click the download link of the template you want to download.

The available download links are:

**Download Excel report template** You receive an Excel file with a sheet that contains the data. You can specify additional information in adjoining sheets, for example diagrams.

**Download BIRT report template** You receive the report data as an empty generic BIRT report template. The datasource is already configured.

**Download as CSV** You receive the report data as a Comma Separated Values (CSV) file. Depending on your local settings, you will receive ',' or ';' as the delimiter character. The date is also formatted based on user settings.

**Download as XML** You receive the report data as XML. The advantage of this approach over CSV is that you retain all sub-report data. Accessing data outside of Performance Manager - You can call a specific URL that offers the report data using the following format:

```
http://server/servicesExchange?  
hid=reportData&userName=<username>  
&passWord=<password>&reportFilterID=<ID of the  
report>&type=<csv|xml>
```

5. The **File Download** dialog box displays. Click **Save** and download the report file to your local system as a `rptdesign` or `xls` file, depending on the report type that you are downloading.
6. Edit the report based on your needs using either the BIRT RCP Designer for `rptdesign` files or Excel for `xls` files.

# Customizing Report Templates

With BIRT RCP Designer (BIRT) and Microsoft Excel, you can customize the pre-installed report templates of Performance Manager and create custom report templates. For details on using BIRT, see the *Silk Performance Manager Administration Help* and the BIRT RCP Designer documentation.

To download an existing template for editing:

1. In the menu, click **Performance Manager > Reports**.
2. Select a report that utilizes the BIRT Report Template.
3. Click the **Properties** tab.
4. Click **Download Excel report template** or **Download BIRT report template**. Depending on your selection, you receive the report data as an empty generic BIRT or Excel report template. The datasource is already configured.
5. Once you have saved the template to your local system, modify it as required.

For detailed information on configuring BIRT report templates, see the *Silk Performance Manager Administration Help*.

6. To upload the modified report template, click **Administration > Reports** in the menu and click **Upload**.

## Removing Report Templates

To remove the template of the current report:

1. In the menu, click **Performance Manager > Reports**.
2. In the **Reports** tree, select the report from which you want to delete a template.
3. Click the **Report** tab.
4. Click .
5. Click **Yes** on the subsequent confirmation dialog box.

## Editing Report Properties

The basic properties of each report are listed on the report's **Properties** tab. To edit the properties of a report:

1. In the menu, click **Performance Manager > Reports**.
2. Select a report in the **Reports** tree.
3. On the **Properties** tab, click **Edit**.
4. On the **Edit Report** dialog, modify the **Name** and **Description** of the report as required.
5. Ensure that the **Share this report with other users** check box is selected if you intend to have this report shared with other users.
6. Depending on whether the selected report is "simple" or "advanced", you may do one of the following:
  - Simple report: You can modify the selection criteria-and thus changing the results of the selected report-or you can click the **Advanced Query** button to modify the SQL query code.
  - Advanced report: If you have familiarity with SQL, you may edit the query code in the **Report data query** field. To assist you in editing SQL queries, a list of function placeholders (i.e., "variables") is available. To insert one of the available pre-defined functions, select the corresponding placeholder from the **Insert placeholder** list.



**Note:** If you manually edit the SQL code for the query, it is recommended that, once complete, you click the **Check SQL** button to confirm your work.

See [Creating New Reports](#) for detailed information about selection criteria and modifying SQL queries.

7. Once you have completed editing the report's properties, click **Finish** to save your settings.

## Editing Report Parameters

Parameters are customizable statement elements. Parameters can be defined any time before a report execution by simply changing them in the Parameters tab.

The **Parameters** page lists customizable statement elements. Parameters can be defined any time before a report execution by simply changing them on the **Parameters** page. The syntax of a parameter is: `$(parametername|defaultvalue|guiname)`. The `defaultvalue` and the `guiname` are optional. Parameter-names cannot contain whitespace characters.

When a report has parameters associated with it, it is possible to edit the values of the parameters before each report execution. Parameter values are stored in the current user context, which means edited values are available only to the user who performs the edits. When parameter values are not specified for a given report execution, the default values from the report definition are used.

When a report has subreports assigned to it, the parameters of those subreports are also shown in the **Parameters** page and the values are stored only within the context of the selected report. For example, the values are only used in conjunction with the current subreport configuration. When creating new reports, parameters are the values that are defined on the **Create New Report** dialog box in the **Selection criteria** area. See [Creating New Reports](#) for details.

To edit the parameters of a report:

1. In the menu, click **Performance Manager > Reports**.
  2. Select a report in the **Reports** tree.
  3. Click the **Parameters** tab. If the report has parameters defined for it, the parameters are listed here.
  4. Click **Edit Parameters**. The **Edit Parameters** dialog box appears.
  5. Edit the **Label** or **Value** of the listed parameters as required.
  6. From the **Usage** list, select the usage type of the parameter:
    - Constant Value
    - Start Time
    - End Time
- Start Time** and **End Time** are used for reports that query for a specific date range.
7. Click **OK**.

## Working with Charts

The **Chart** page enables you to define charts and graphs for data analysis.

The page relies on the internal reporting engine of Performance Manager to create standard charts and graphs from the data retrieved by the selected report query.

## Displaying Charts

To display a chart:

1. In the menu, click **Performance Manager > Reports**.
2. Select a report in the **Reports** tree.
3. Click the **Chart** tab to display the default chart.
4. To select a different chart type, click . The **Select Chart Type** dialog appears.
5. Select a chart type from the **Chart type** list.
6. Check the view properties that you want to apply to the chart:
  - 3D view
  - Show horizontal grid lines
  - Show vertical grid lines
  - Show legend
7. Specify how these chart options are to be saved:
  - Click the **For current user only** option to have these chart settings override the report's standard settings whenever the current user views this chart.
  - Click the **As report standard** option to have these chart settings presented to all users who do not have overriding user settings defined. This setting does not effect individual user settings.
8. Click **OK** to display the new chart type.



**Note:** The chart configurations you define here become the default for this report. When standard charts and graphs are not able to deliver the specific data that you require, or when they cannot display data in a required format, you can customize the appearance of queried data using the Performance Manager reporting functionality. To open the current chart in a separate browser window, click  at the top of the **Chart** page.

## Printing Charts

To print the current chart:

1. In the menu, click **Performance Manager > Reports**.
2. Select a report in the **Reports** tree.
3. Click the **Chart** tab.
4. Click the **Print** icon. The chart data displays in a new window in printable format. Your system's print dialog box is also displayed.
5. Configure print settings as necessary and click **OK** to print the chart.

## Removing Charts

Removing a chart only removes the currently selected chart template from the selected report, it does not remove the chart template entirely.

To remove the current chart template from the selected report:

1. In the menu, click **Performance Manager > Reports**.
2. Select a report in the **Reports** tree.
3. Click the **Chart** tab.
4. Click  (**Remove chart type**). The **Remove Chart** dialog box opens.
5. Select one of the following:
  - Select **Remove user settings (and revert to report standard)** to have the current user's chart settings deleted along with the chart. The chart will subsequently be displayed according to the report's standard settings. If no standard settings have been defined, the chart cannot be displayed.  
 **Note:** This option is only available when the current user has defined specific chart settings.
  - Select **Remove standard chart settings** of report to have any standard settings deleted along with the chart. User-specific settings are not affected by this option.  
 **Note:** This option is only available when standard chart settings have been defined for a report.
6. Click **OK** to delete the chart template. If required, you can click the **<Click here to choose a chart type>** link to assign a new chart template to the selected report.

## Viewing Reports

Because each template expects a certain data format to produce a useful graph, not all templates can be applied to all report queries. You will receive an error message if you attempt to generate a report through an incompatible report template.

To generate a report:

1. In the menu, click **Performance Manager > Reports**.
2. In the **Reports** tree, select the report that you want to generate.

3. Click the **Report** tab.
4. Click the **Select Report Template** icon. The **Select Report Template** dialog box displays.
5. Select the template you wish to use.
6. Click **OK** to display the report.

## Saving Reports

How you save a report locally depends on whether you have selected a BIRT report template or an Excel or Word template. If you have selected an Excel template, click the **Download Excel report template** link on the **Properties** page of the selected report. This will invoke Microsoft Excel on your local computer and the report will be loaded automatically.

If you have selected a BIRT report template, use the following procedure to save the report as PDF:

1. In the menu, click **Performance Manager > Reports**.
2. In the **Reports** tree, select the report that you want to save.
3. Click the **Report** tab.
4. Click  on the **Report view** toolbar.
5. On the **File Download** dialog box, click **Save** to save the PDF document to a location of your choice.

# Index

## A

- accuracy
  - project overview report 45
- action Essentials
  - writing 63
- adding
  - custom incidents 34
  - custom monitor schedules 23
  - definite runs 26
  - exclusions 25
  - monitors 15
  - rules 29
  - sub-reports 76
  - transaction conditions 27
- Administrator
  - user roles 9
- Analyst
  - user roles 9
- analyzing
  - health 39
- API
  - web services 11
- availability
  - project overview report 44

## B

- BIRT
  - customizing report templates 77
- boundaries
  - configuring 21

## C

- calculation
  - performance rate 62
- charts
  - displaying 79
  - printing 80
  - removing 80
- client monitors
  - overview 13
- concepts
  - Performance Manager 7
- configuring
  - boundaries 21
  - infrastructure monitors 59
  - project schedules 23
  - rule actions 32
- correlating
  - results 50
- creating
  - infrastructure monitors 60
  - reports 72
  - Silk Test monitors 19
- custom incidents
  - adding 34

- deleting 35
  - editing 35
  - overview 33
- custom monitor schedules
  - adding 23
  - deleting 25
  - editing 24
- custom reports
  - SQL functions 73
- customizing
  - start page 42
- customizing report templates
  - BIRT 77
  - Excel 77

## D

- definite runs
  - adding 26
  - deleting 27
  - editing 27
- deleting
  - custom incidents 35
  - custom monitor schedules 25
  - definite runs 27
  - exclusions 26
  - monitors 17
  - rules 32
  - sub-reports 76
  - transaction conditions 28
- dimensions
  - health rates 66
- displaying
  - charts 79
- distribution
  - monitors 9
- downloading
  - report templates 77

## E

- editing
  - custom incidents 35
  - custom monitor schedules 24
  - definite runs 27
  - exclusions 26
  - monitors 17
  - report parameters 78
  - report properties 78
  - rules 31
  - transaction conditions 28
- emailing
  - reports 43
- Essentials
  - overview 8
- Excel
  - customizing report templates 77
- exclusions

- adding 25
- deleting 26
- editing 26
- execution
  - log 56
- execution servers
  - starting as Windows process 20
- exporting
  - Silk Test projects 19

## G

- GUI-level testing
  - execution server configuration 20

## H

- health
  - analyzing 39
- health detail reports
  - heat fields 40
- health rates
  - accuracy 66
  - availability 66
  - dimensions 37, 66
  - overview 65
  - performance 66
- heat fields
  - health details 40

## I

- incidents
  - log 54
- infrastructure monitors
  - configuring 59
  - creating 60

## L

- log
  - incidents 54
  - service target 55

## M

- maintaining
  - monitors 61
- monitoring
  - windows machines 59
- monitors
  - adding 15
  - deleting 17
  - distribution 9
  - editing 17
  - maintaining 61
  - pre-installed 8
  - reusability 58
- monitors and transactions
  - relationship 14

## O

- overall health 71

## P

- performance
  - detail charts 48
- Performance Manager
  - concepts 7
  - product overview 6
- performance rate
  - calculation 62
- pre-installed
  - monitors 8
- prerequisites
  - Silk Test monitors 18
- printing
  - charts 80
- product overview
  - Performance Manager 6
- Project Manager
  - user roles 9
- project overview report
  - accuracy 45
  - availability 44
  - details 41
  - performance 47
- project owner
  - contacting 44
- project schedules
  - configuring 23
  - overview 22

## R

- recovery system
  - Silk Test 18
- removing
  - charts 80
  - report templates 78
- report parameters
  - editing 78
- report properties
  - editing 78
- report templates
  - downloading 77
  - overview 76
  - removing 78
  - uploading 76
- Reporter
  - user roles 9
- reports
  - chart page 79
  - creating 72
  - deleting sub-reports 76
  - displaying charts 79
  - editing parameters 78
  - editing properties 78
  - emailing 43
  - generating 80
  - overview 72
  - printing charts 80
  - removing charts 80
  - removing templates 78

- saving 81
- uploading templates 76
- viewing 80
- writing advanced SQL queries 74

result files

- writing 63

results

- correlating 50
- root cause analysis 52

root cause

- analyzing 52

rule actions

- configuring 32

rules

- adding 29
- deleting 32
- editing 31
- overview 29

## S

saving

- reports 81

scheduling

- Silk Test monitors 20

service manager

- starting execution server as Windows process 20

service target

- log 55

setting up

- infrastructure monitors 60

Silk Performer

- working with 10

Silk Test monitors

- creating 19
- prerequisites 18
- scheduling 20

Silk Test projects

- exporting 19

SQL functions

- reports 73

SQL reports

- example 75

start page

- customizing 42

sub-reports

- adding 76
- deleting from reports 76
- overview 76

SuperUser

- user roles 9

## T

time zones

- overview 22

transaction conditions

- adding 27
- deleting 28
- editing 28
- overview 27

TrueLog files

- downloading 54

## U

uploading

- report templates 76

user roles

- description 9

## W

web services

- API 11

windows machines

- monitoring requirements 59

working with

- Silk Performer 10

writing

- action Essentials 63
- result files 63